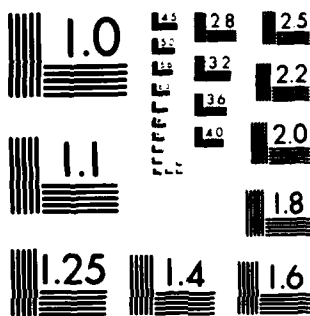


UNCLASSIFIED

1/1

NL

END
DATE
FILMED
1. - 62
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

DOT/FAA/CT-82/89

Remote Maintenance Monitoring System Concentrator

AD A127133

David Wainland

Prepared By
FAA Technical Center
Atlantic City Airport, N.J. 08405

December 1982

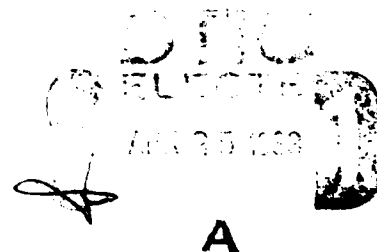
Final Report

This document is available to the U.S. public
through the National Technical Information
Service, Springfield, Virginia 22161.

DTIC FILE COPY



US Department of Transportation
Federal Aviation Administration
Airway Facilities Service
Washington, D.C. 20590



NOTICE

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for the contents or use thereof.

The United States Government does not endorse products or manufacturers. Trade or manufacturer's names appear herein solely because they are considered essential to the object of this report.

Technical Report Documentation Page

1. Report No. DOT/FAA/CT-82/89		2. Government Accession No. AD-A127133		3. Recipient's Catalog No.	
4. Title and Subtitle REMOTE MAINTENANCE MONITORING SYSTEM CONCENTRATOR				5. Report Date December 1982	
				6. Performing Organization Code ACT-100	
7. Author(s) David Wainland				8. Performing Organization Report No. DOT/FAA/CT-82/89 ✓	
9. Performing Organization Name and Address Federal Aviation Administration Technical Center Atlantic City Airport, New Jersey 08405				10. Work Unit No. (TRAIS)	
				11. Contract or Grant No. 144-170-900	
12. Sponsoring Agency Name and Address U.S. Department of Transportation Federal Aviation Administration Airway Facilities Service Washington, D.C. 20590				13. Type of Report and Period Covered Final Report April 1981 - June 1982	
				14. Sponsoring Agency Code	
15. Supplementary Notes					
16. Abstract A Remote Maintenance Monitoring System (RMMS) concentrator has been designed, developed, and tested at the Federal Aviation Administration (FAA) Technical Center. The concentrator is a microcomputer-based device that collects, stores, displays, and retransmits to a Maintenance Processor Subsystem, performance information obtained from many remote navigational aid monitors. The concentrator consists of a communications subsystem and a data subsystem. Due to its design features, the concentrator may be reconfigured to handle several RMMS tasks. By incorporating operating system software into the data subsystem, the concentrator becomes a low cost, general purpose minicomputer.					
17. Key Words Remote Maintenance Monitoring System, RMMS Concentrator, Microcomputer, Communications and Data Concentrator, Remote Monitoring Subsystem Processor (RMSP)			18. Distribution Statement This document is available to the U.S. public through the National Technical Information Service, Springfield, Virginia 22161		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 56	
				22. Price	

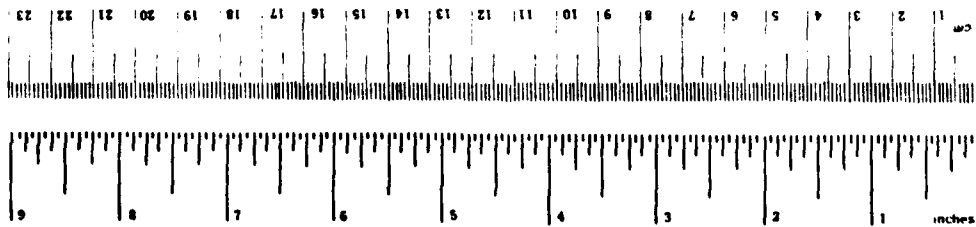
METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

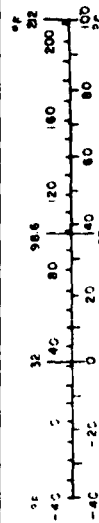
Symbol	When You Know	Multiply by	To Find	Symbol
LENGTH				
in	inches	2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
AREA				
sq in	square inches	6.5	square centimeters	cm ²
sq ft	square feet	0.09	square meters	m ²
sq yd	square yards	0.8	square meters	m ²
sq mi	square miles	2.6	square kilometers	km ²
	acres	0.4	hectares	ha
MASS (weight)				
oz	ounces	28	grams	g
lb	pounds	0.46	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
VOLUME				
teaspoon	teaspoons	5	milliliters	ml
tablespoon	tablespoons	15	milliliters	ml
fluid ounce	fluid ounces	30	milliliters	ml
cup	cups	0.24	liters	l
pint	pints	0.47	liters	l
quart	quarts	0.96	liters	l
gallon	gallons	3.8	liters	l
cubic foot	cubic feet	0.03	cubic meters	m ³
cubic yard	cubic yards	0.76	cubic meters	m ³
TEMPERATURE (exact)				
F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	C

Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
LENGTH				
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
AREA				
cm ²	square centimeters	0.16	square inches	in ²
m ²	square meters	1.2	square yards	yd ²
km ²	square kilometers	0.4	square miles	mi ²
ha	hectares (10,000 m ²)	2.5	acres	ac
MASS (weight)				
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
VOLUME				
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m ³	cubic meters	35	cubic feet	ft ³
m ³	cubic meters	1.3	cubic yards	yd ³
TEMPERATURE (exact)				
C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	F



Scale is 2.54 inches. For conversion from centimeters to inches, divide by 2.54. For conversion from inches to centimeters, multiply by 2.54.



FOREWORD

When Task 9 of 9550-AAF-501-78-002 was initially assigned to the Federal Aviation Administration (FAA) Technical Center, the name "concentrator" was used in the task assignment to define the Remote Maintenance Monitoring System (RMMS) concentrator covered in this report. Consequently, during the performance of this task, the name concentrator has been used in all documentation generated by the Technical Center to preserve continuity.

The official name for the concentrator has now been changed to the Remote Monitoring Subsystem Processor (RMSP). Therefore, whenever the word concentrator is used in this report, it is the same as the RMSP.



TABLE OF CONTENTS

	Page
INTRODUCTION	1
Purpose	1
Background	1
CONCENTRATOR DESIGN	3
Design Philosophy	3
Concentrator Overview	3
Communications Subsystem	5
Data Subsystem	12
REMOTE MONITOR SIMULATORS	19
Remote Monitor Simulator Tasks	21
Hardware Design	21
Simulator Software	22
MAINTENANCE PROCESSOR SUBSYSTEM SIMULATOR	27
CONCENTRATOR TESTING	31
ANALYSIS OF CONCENTRATOR	34
System Flexibility	34
Software Analysis	38
CONCLUSIONS	40
RECOMMENDATIONS	41
REFERENCES	42
APPENDICES	
A — Formatted Data Information	
B — System Photographs	

LIST OF ILLUSTRATIONS

Figure		Page
1	Remote Maintenance Monitoring System	1
2	Concentrator Block Diagram	4
3	Communications Subsystem Local Terminal Commands	7
4	Communications Subsystem	8
5	Communications Subsystem Main Routine Flowchart	10
6	Data Subsystem Block Diagram	14
7	Data Subsystem Main Routine Flowchart	16
8	Comparison of Raw and Formatted DME Data	18
9	Communications Messages on Data Subsystem Local Terminal	18
10	Alarm Messages on Data Subsystem Local Terminal	18
11	Message from the MPS Simulator Displayed on Data Subsystem Terminal	19
12	Data Subsystem Local Terminal Commands	20
13	Z-80 Remote Monitor Simulator Block Diagram	23
14	Twenty-four Channel Remote Monitor Simulator Block Diagram	24
15	Single-Channel Simulator Main Routine Flowchart	25
16	Remote Monitor Simulator Local Terminal Command	28
17	MPS Simulator Block Diagram	29
18	MPS Simulator Main Routine Flowchart	30
19	Concentrator Test Configuration	32
20	Communications and Data Subsystems Showing Multibus Chains	37
21	Communications and Format/Display System	39

LIST OF TABLES

Table		Page
1	Types of Central Buffers	11
2	Summary of Concentrator Options	35
3	Cost Breakdown for the Technical Center's Concentrator	35

INTRODUCTION

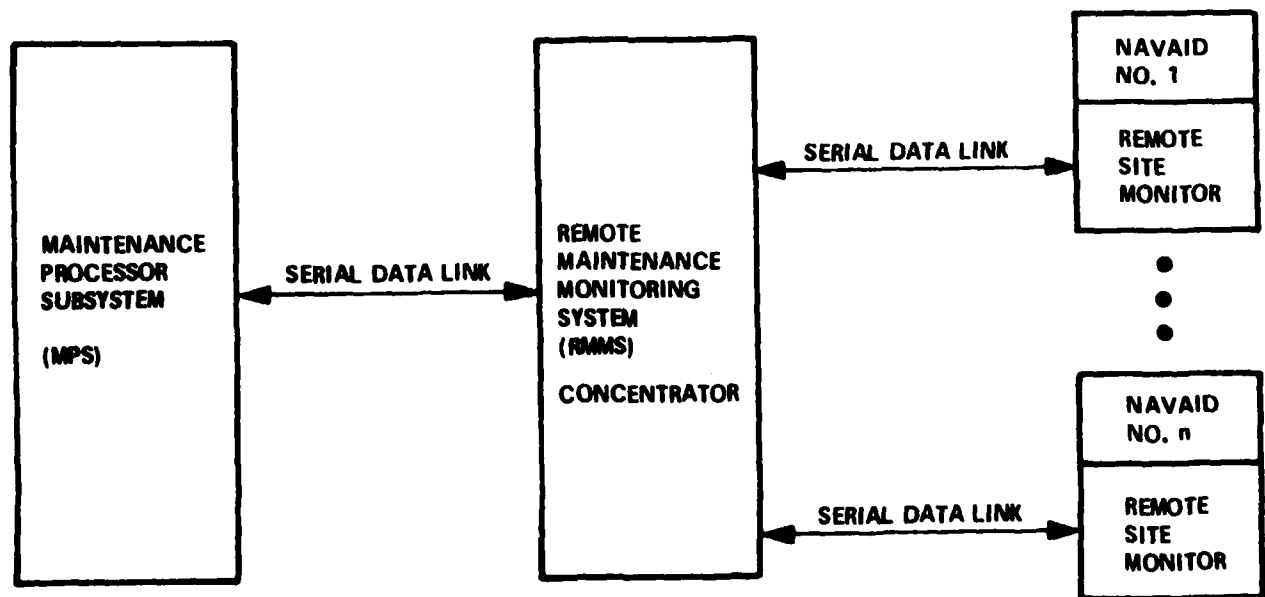
PURPOSE.

As part of Task 9 of the Interservice Agreement (9550-AAF-501-78-002) for the Remote Maintenance Monitoring System (RMMS) Program, the Federal Aviation Administration (FAA) Technical Center designed and developed an RMMS concentrator. This report will describe the concentrator, several associated simulation devices, and the tests conducted to verify the concentrator performance.

BACKGROUND.

The FAA has the responsibility of maintaining the navigational aids (NAVAID's) used in the National Airspace System. (The word NAVAID's is used throughout this report in the broad sense to include radar, beacon, landing systems, etc.) These NAVAID's are dispersed throughout the United States. To provide more efficient utilization of available resources, the FAA is implementing a plan that calls for the consolidation of Airway Facilities Sector work centers, the purchase of more reliable and maintainable NAVAID's, and remote maintenance monitoring (RMM) of the NAVAID's.

The Technical Center was responsible for developing, documenting, and testing a feasibility concentrator model. It was proposed that this concentrator be incorporated into the RMM system as shown in figure 1.



82-89-1

FIGURE 1. REMOTE MAINTENANCE MONITORING SYSTEM

System operation would occur as follows:

1. At each NAVAID a computer, called a Remote Monitoring Subsystem (RMS), interfaces to the NAVAID via a set of sensors.
2. The computer monitors the performance of the NAVAID.
3. The performance information from each remote monitor is sent to a concentrator (also a computer system) via serial data channels (i.e., telephone lines, microwave links, etc.).
4. The information collected by each concentrator is sent to a high level computer called the Maintenance Processor Subsystem (MPS).
5. The MPS, which will probably be located at Airway Facilities Sector work centers, stores and displays the collected information.

Task 9 did not place any constraints on the concentrator design other than requiring the communications characteristics to be consistent with references 1 and 2. Task 9 did not specify or limit the functions to be performed by the concentrator. In addition, specific requirements (such as the number of remote monitors to be serviced by each concentrator, the data rates between a remote site and concentrator, the geographic locations of the concentrators, etc.) were not stated.

For these reasons, a determination of the concentrator's functions and requirements was made at the Technical Center. It was decided that the Technical Center's concentrator should be designed to perform the following three functions:

1. Communicate with remote monitors and an MPS.
2. Store the information obtained from the remote monitors.
3. Format and display the collected information.

In making this decision, it was realized that many concentrator applications may not require information storage or display. Also, an effort had to be made to minimize concentrator cost. Therefore, a flexible, modular, and expandable concentrator design was required.

This modular approach would make it possible to use only those subsystems needed to perform the desired function(s). For example, if the information storage and display functions are not needed, the concentrator would contain only hardware and software to support the communication function. In this way, cost could be minimized.

In the remaining sections of this report, the design of the concentrator, the remote monitor simulators, and an MPS simulator will be discussed. Procedures for test and evaluation of the concentrator will also be discussed. Recommendations regarding potential applications of the concentrator will then be made.

CONCENTRATOR DESIGN

DESIGN PHILOSOPHY.

As discussed in the "Introduction," the concentrator design had to be flexible, modular, and expandable. To meet these requirements, it was decided to use the Multibus™ bus structure and the 8086 microprocessor.

Multibus was developed by the Intel Corporation to provide a means of multi-processing. Using the Multibus structure, it is possible to have several processor boards share the same card cage, common memory, and common input/output (I/O) devices. By taking advantage of this capability, if a single processor is not fast enough to perform the required tasks, a second processor can be added. Individual processors can be programmed to handle special tasks. For example, in the concentrator application, one processor might be dedicated exclusively to handling communications while a second processor simultaneously performs information storage and display functions. Both processors could share the same card cage and memory. Also, different types of processors can be used as long as they are on Multibus compatible circuit boards.

There are over 40 vendors that are providing Multibus compatible boards. As a result, a wide range of computer boards, memory boards, and I/O boards are available.

The 8086 microprocessor was chosen for the following reasons:

1. An 8086 hardware/software development facility was available at the Technical Center.
2. Processor speed is required in handling communications interrupts. The sixteen-bit 8086 is one of the fastest microprocessors available.
3. The concentrator has a requirement for storing large data buffers. The 8086 can handle this requirement since it can address up to a megabyte of memory.
4. Multibus compatible 8086 computer boards are available from several vendors.

Multibus and the 8086 were chosen as they guarantee sufficient processing power and the flexibility needed for this design. The concentrator could have been designed using an off-the-shelf minicomputer system, however, this would have been more expensive than the approach taken. Furthermore, the advantage of having many vendors making compatible products would not exist.

CONCENTRATOR OVERVIEW.

A block diagram of the concentrator is shown in figure 2. The concentrator consists of a communications subsystem and a data subsystem. The communications subsystem transfers information to and from remote site computers via serial data channels (or ports). Information collected by the communications subsystem is transferred to the data subsystem, also using a serial data channel. The data

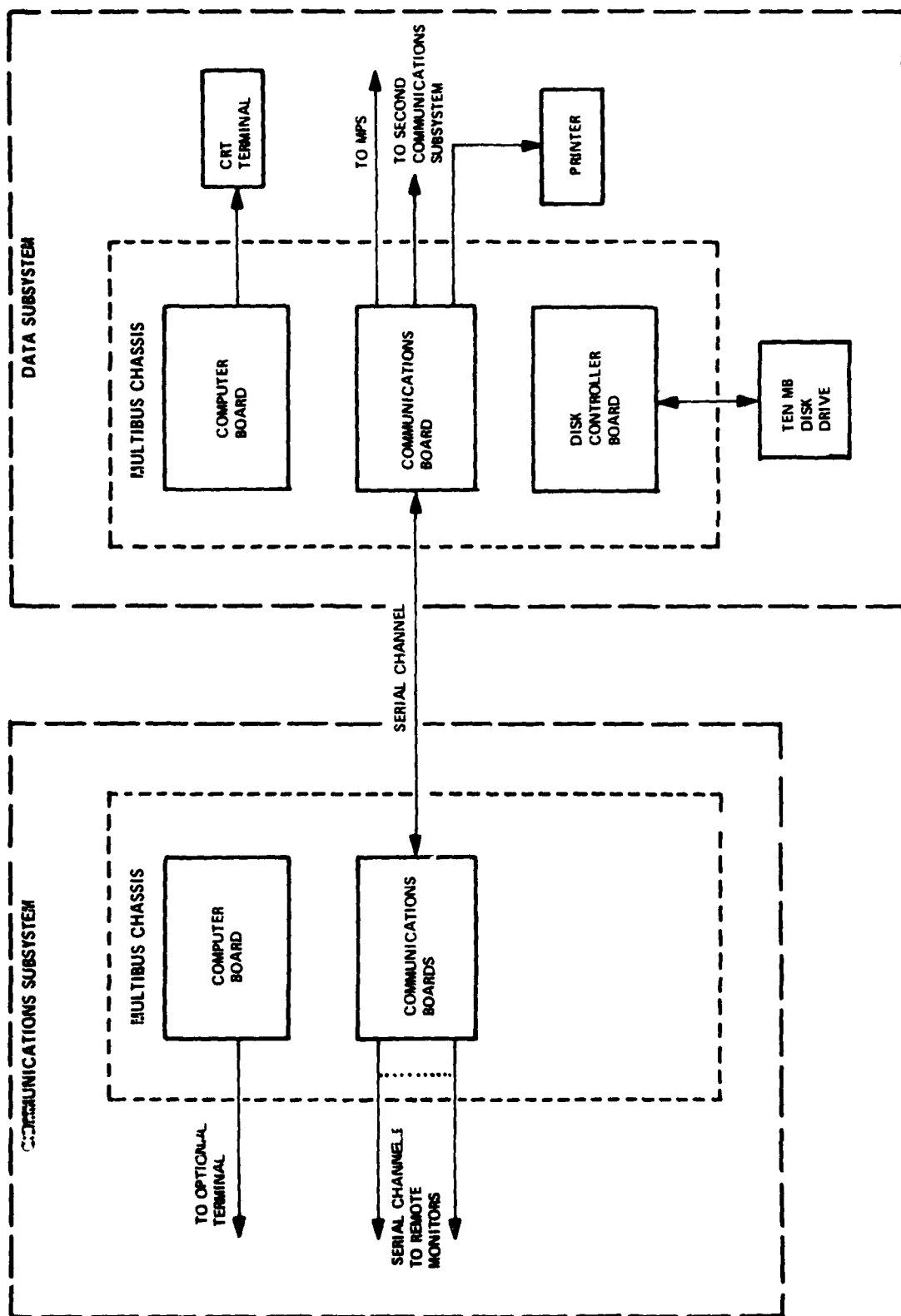


FIGURE 2. CONCENTRATOR BLOCK DIAGRAM

subsystem stores, formats, and displays the information. The data subsystem also transfers the information to an MPS. The two subsystems need not be collocated. Appendix B (figure B-1) contains a photograph of the concentrator built at the Technical Center.

The communications and data subsystems will be described in the following sections. The hardware and software designs are provided for each subsystem description of the tasks performed.

This section will describe both subsystems in detail. For each subsystem, a description of the tasks performed and the hardware/software designs will be provided.

COMMUNICATIONS SUBSYSTEM.

COMMUNICATIONS SUBSYSTEM TASKS. The purposes of the communications subsystem are to provide continuous communication with many remote monitors, and to send the information obtained from the remote monitors to the data subsystem or, in some cases, to the MPS.

The communications subsystem performs the following tasks:

1. It institutes general polls (called continuous polls in reference 1) of each remote monitor to obtain the associated NAVAID's status and any alarm data or messages that have occurred since the last general poll. General polls are sent to each site every 3 seconds. However, by making the proper keyboard entries at the terminal, the general poll rate can be varied from 1 to 9 seconds.
2. It performs periodic polls (called scheduled polls in reference 1) of all remote monitors.
3. It performs error checking on all information received from the remote monitors and the data subsystem.
4. It provides the data subsystem with the following data in the following order of priority:
 - a. All alarm data.
 - b. All periodically polled data and all data polled manually from the data subsystem terminal (reference 1 calls these scheduled polls).
 - c. All messages from remote sites addressed to the data subsystem.
 - d. All requests made at one site for data at a different remote site.
5. It informs the data subsystem when a loss in communications with a remote monitor occurs, and also informs the data subsystem when communications with a remote monitor is resumed.
6. It routes messages from a terminal at one remote monitor or the data subsystem to the terminal at the remote monitor addressed by the message.

7. Via keyboard entries, the following can be accomplished:

- a. The number of channels polled can be reduced from the maximum of 35.
- b. Periodic polls can be varied between 1 and 99 hours.
- c. General polls can be varied between 1 and 9 seconds.
- d. Data rates can be varied between 110 and 1200 bits per second to the remote monitor.
- e. Time and date entries can be made.

Examples of communications subsystem terminal operations are presented in figure 3.

It should be noted that the communications protocol used is not completely compatible with the Interface Communications Document, Level 1 (ICD-1) (reference 1). At the time this development effort began, ICD-1 was being revised. For this reason, the protocol used in the concentrator was developed at the Technical Center. However, this protocol is similar to the asynchronous byte-oriented protocol in the ICD-1.

HARDWARE DESIGN. The communications subsystem consists of an Intel SBC-660 Multibus chassis, an Intel SBC-86/12A microcomputer board, an Intel SBC-300 memory multimodule, an Intel SBC-534 four-channel serial communications board, four Central Data B1018 eight-channel serial communications board, and a portable cathode ray tube (CRT) terminal from Applied/Digital Data Systems. The subsystem block diagram is presented in figure 4.

The SBC-86/12A contains the 8086 microprocessor, 32K bytes of random access memory (RAM), and the capability to accommodate up to 16K bytes of programmable read only memory (PROM). In addition, the board contains a priority interrupt controller, a serial data channel, a parallel interface circuit, a system clock, and a programmable interval timer. The serial data channel is connected to the CRT terminal.

The SBC-300 memory multimodule contains 32K bytes of RAM. It physically and electrically interfaces to the SBC-86/12A computer board. This module expands the subsystem's RAM to 64K.

The SBC-534 board contains four serial data channels, auto-dial circuitry, a programmable interval timer, a priority interrupt controller, and a clock circuit. One serial channel handles communications with the data subsystem. The other three channels communicate with remote monitors.

Each B1018 board contains eight serial data channels and a clock circuit. The channels are all used to communicate with remote monitors.

The serial channels on all the boards are capable of both asynchronous and synchronous communications at data rates up to 19200 bits per second (bps). The software design used constrains the system to asynchronous communications at data rates up to 1200 bps. The decision to limit the data rate and to provide a maximum of 35 channels to remote monitors is based on a study described in reference 3. However, both the number of channels and the data rates can be increased if it becomes necessary.

H
TYPE ONE OF THE FOLLOWING:
B-SELECT BAUD RATE
C-SELECT NO. OF CHANNELS
D-SET DATE
G-SET GENERAL POLL TIMER
F-SET PERIODIC POLL TIMER
T-SET TIME

B WHICH CHANNEL? 7 TYPE 1 FOR 110, 2 FOR 300, 3 FOR 1200 2 OK

C ENTER LAST CHANNEL ADDRESS- S OK

D ENTER DATE- 03/25/82. OK

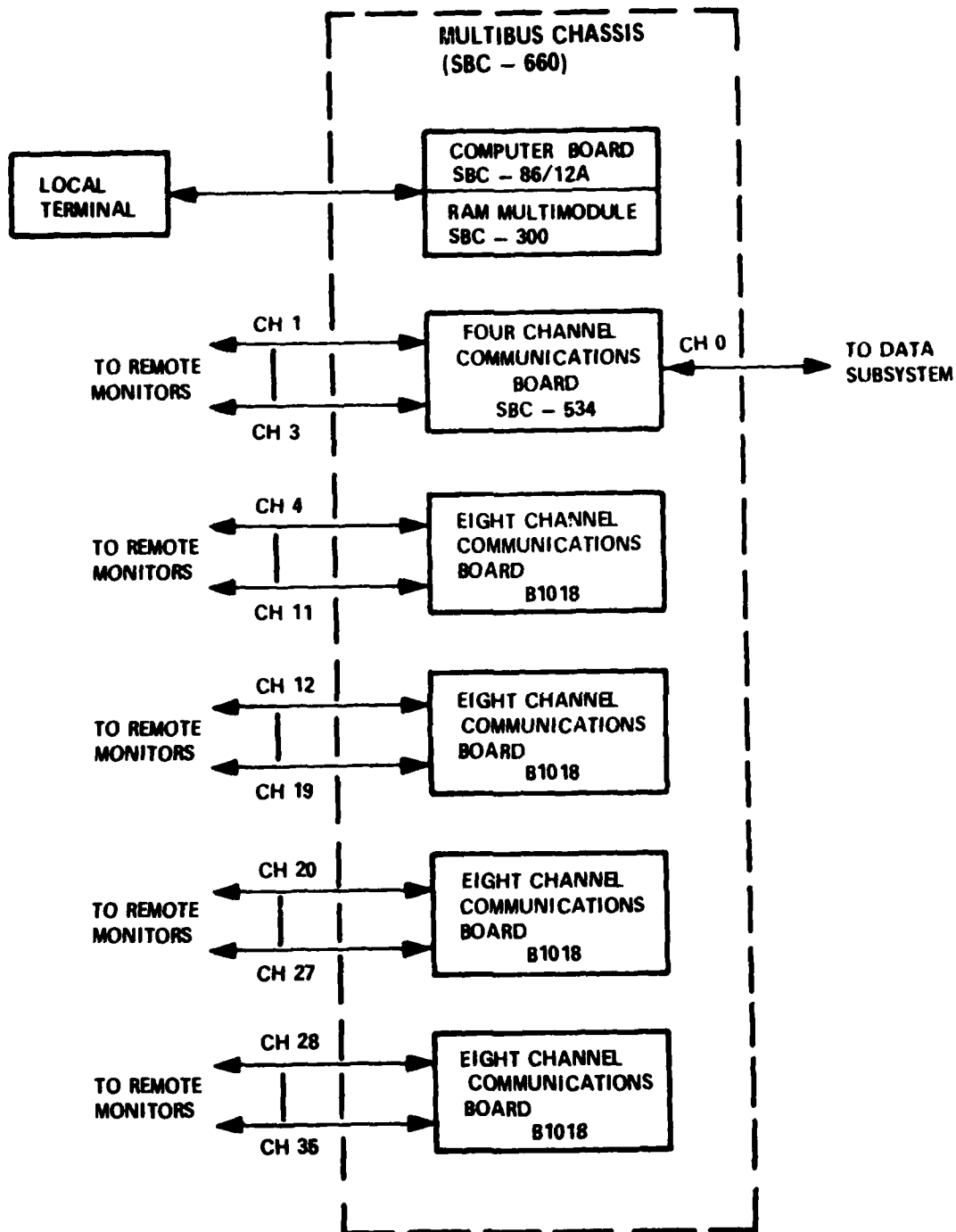
G ENTER GENERAL POLL TIMER IN SECONDS- 3 OK

F WHICH CHANNEL? A ENTER PERIODIC POLL TIMER IN HOURS- 12. OK

T ENTER TIME- 09:30:00. OK

82-89-3

FIGURE 3. COMMUNICATIONS SUBSYSTEM LOCAL TERMINAL COMMANDS



82-89-4

FIGURE 4. COMMUNICATIONS SUBSYSTEM

The CRT terminal used is RS-232 compatible. Any RS-232 compatible terminal may be used. The terminal is not required for subsystem operation, but, when used, it provides additional capabilities for the subsystem (i.e., polling rates and data rates can be changed, etc., as shown in figure 3.)

The SBC-660 chassis contains an eight-slot card cage, a power supply containing ± 12 volts and ± 5 volts, fans for cooling, and a front panel with power and reset switches.

SOFTWARE. The communications subsystem software consists of: initialization, main, receive interrupt, terminal interrupt, and real-time clock interrupt routines. All software was written in PLM-86, a high level structured language for the 8086 microprocessor. This subsystem's software and all other 8086 microprocessor based software described in this report were developed on an Intel MDS-230 development system. Software/hardware interfacing and debugging was accomplished using an Intel ICE-86 in-circuit emulator connected to the MDS-230.

The executable code required approximately 8.5K bytes of PROM. This was accomplished using the PLM-86 compiler's highest level of optimization. Without optimization, the amount of PROM required would increase, and so would the time required for program execution. About 40K bytes of RAM were required for data storage.

Operation of the subsystem software is as follows:

1. Subsystem operation begins by depressing the reset button on the SBC-660 front panel. The computer immediately begins executing the initialization routine. The initialization routine resets all storage areas and initializes all program-mable I/O devices on the microcomputer board and the communications boards.
2. Program execution transfers to the main routine. The main routine consists of a continuous loop that calls several subroutines. A simplified flowchart of the main routine is shown in figure 5.
3. The first subroutine called is MOVBLK. This subroutine is executed once for each channel and determines if a complete block of information has been received from a remote channel or the data subsystem. If a complete block has not been received, execution returns to the main routine. Otherwise, MOVBLK transfers the information from the receive buffer to one of several central buffers. Which central buffer is selected depends on the type of information just received. For example, alarm data are transferred to the central alarm buffer, periodic data to the central periodic buffer, etc. The types of central buffers are summarized in table 1. After moving the information, execution returns to the main routine.
4. TRNMIT is called next. This subroutine is called once for each serial channel to a remote monitor, and once to service the channel to the data subsystem. If a channel transmit buffer contains data, TRNMIT determines if the serial channel is ready to transmit the next byte of that data. If the channel is ready, the next byte of information is supplied and then program execution returns to the main routine. If the channel is not ready to transmit, program execution immediately returns to the main routine.

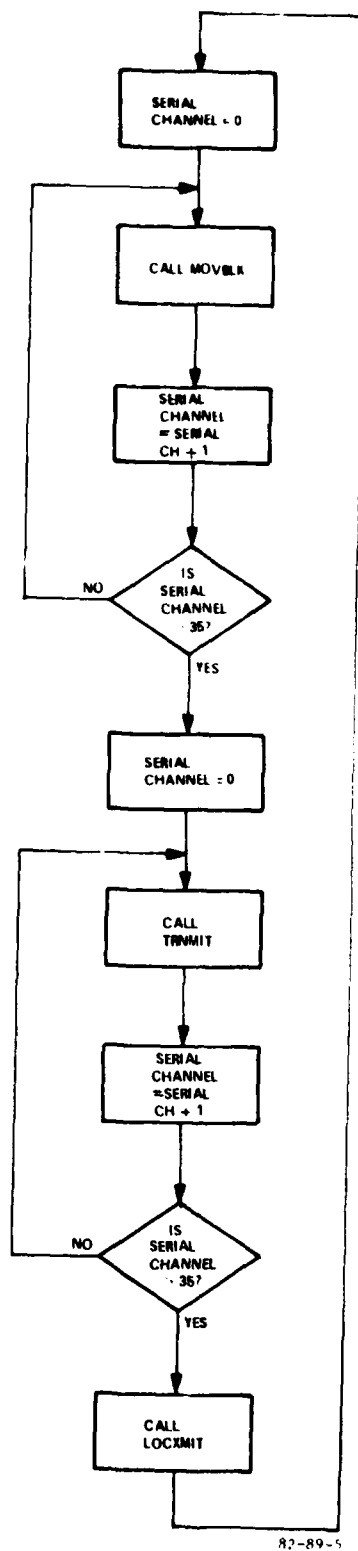


FIGURE 5. COMMUNICATIONS SUBSYSTEM MAIN ROUTINE FLOWCHART

TABLE 1. TYPES OF CENTRAL BUFFERS

<u>Buffer Type</u>	<u>No. of Blocks</u>	<u>Block Size (bytes)</u>	<u>Total Amount of Storage (bytes)</u>
Alarm Data from Remote Monitor	36	256	9216
Periodic Data from Remote Monitor	16	256	4096
Message from Remote Monitor to Data Subsystem	8	256	2048
Message to a Remote Monitor	16	256	4096
Data Request from a Remote Monitor	8	16	128
Periodic Data Request from Data Subsystem	16	32	512
Communications Status	36	16	576
Total			20,672

If information is not currently being sent to a remote monitor, TRNMIT determines if information should be sent and what type of information should be sent. For communications with a remote site, this would include general polls, periodic polls, messages, or the acknowledgment (ACK) or negative acknowledgment (NAK) responses to receipt of information from the remote monitor. Once the correct information is obtained, it is placed in that channel's transmit buffer.

Transmissions to the data subsystem are a little more complex. If a message has been received from the data subsystem, TRNMIT will place an ACK or NAK in the transmit buffer going to the data subsystem. (Negative acknowledgment would be required if an error in the received data was detected by the receive interrupt routine.) Otherwise, TRNMIT will begin examining the contents of the central buffers, starting with the central alarm buffers.

The oldest block of alarm information stored in the central buffer will be transferred to the transmit buffer so that it can be sent to the data subsystem. Each time the transmit buffer is emptied, the oldest block of alarm information is transferred to the transmit buffer. Once the central alarm buffers no longer contain alarm information (or if the alarm buffers were all empty to begin with) the oldest block of periodic information in the central periodic buffers is transferred to the transmit buffer. The process described for alarm information is then repeated for the periodic information, messages, etc., until all the information in the central buffers has been sent to the data subsystem. When the central buffers are empty, TRNMIT sends general polls to the data subsystem.

This software sequence insures that alarm information is processed before all other data. It also insures that the first alarm information received by the communications subsystem is the first alarm information sent to the data subsystem.

5. After TRNMIT is executed, LOCXMIT is called. This routine controls the transmission of information to the CRT terminal. If there are data in the transmit buffer that services the terminal and if the serial channel to the terminal is ready to transmit, a character is sent to the CRT and program execution returns to the main routine. If the port was not ready to transmit, program execution immediately returns to the main routine.

6. The process described in steps 3, 4, and 5 is continually repeated. Program execution can be diverted to other routines due to the occurrence of hardware interrupts. Interrupts occur when a character is received on any remote monitor serial channel or the data subsystem channel, or when a keyboard entry is made on the terminal, and every 50 milliseconds when the programmable interval timer on the computer board reaches zero.

a. When a character is received by a serial channel, the receive interrupt routine is entered. This routine first determines which channel received the character. The receive interrupt routine performs error checking on the incoming information and stores the information in that channel's receive buffer. Program execution then returns to the main routine.

b. When a keyboard entry is received by the serial port on the computer board, the terminal interrupt routine is entered. This routine determines what type of request is being made (i.e., request to change polling rates, a time entry, etc.) and how to respond to the request. Responses to the keyboard entries are placed in the terminal transmit buffer and are sent to the terminal under control of the LOCXMIT subroutine described earlier. Keyboard entries and computer responses are shown in figure 3.

c. During the initialization routine, the programmable interval timer on the computer board is programmed to generate an interrupt every 50 milliseconds. When this interrupt occurs, the real-time clock interrupt routine is entered. This routine updates the software timers used to control general and periodic polls. It also updates the ASCII character real-time clock. The ASCII character clock values are transmitted to remote monitors during periodic polls. This routine will also alter the ASCII character clock value if a new value has been received from the terminal or the data subsystem.

DATA SUBSYSTEM.

DATA SUBSYSTEM TASKS. The purposes of the data subsystem are to obtain all the remote monitor information collected by the communications subsystem, to store and display the information, to send the information to an MPS, and to provide a means of manually polling a remote monitor.

The data subsystem performs the following tasks:

1. Receives from the communications subsystem all the alarm information and polled information that the communications subsystem obtained from the remote monitors.

2. Receives via the communications subsystem all messages from the remote monitors that are addressed to the data subsystem.
3. Receives via the communications subsystem all data requests made at one remote monitor for information about the performance of another remote monitor.
4. Receives notification from the communications system when either the loss or resumption of communications to a remote monitor has occurred.
5. Transmits all information it obtained to the MPS.
6. Performs error checking on all information received from the communications subsystem and the MPS.
7. Transmits messages and periodic poll requests made at the data subsystem's CRT terminal to the communications subsystem. (The communications subsystem then transfers the information to the appropriate remote monitor.)
8. Converts data requests made from the terminal at one remote monitor, into a periodic poll addressed to the desired remote monitor. The information returned is formatted and sent to the requesting remote monitor via the communications subsystem.
9. Receives messages from the MPS.
10. Stores all information received on a removable cartridge disk.
11. Formats all alarm data and polled data.
12. Transmits the formatted data and received messages to a line printer.
13. Displays communication status information and alarm warnings on the CRT terminal, when appropriate.

HARDWARE DESIGN. The data subsystem consists of an Intel SBC-660 Multibus chassis, an Intel SBC-86/12A microcomputer board, an Intel SBC-534 four-channel serial communications board, an Xylogics model 410 disk controller board, a Hazeltine 1520 CRT terminal, a Data South DS-180 line printer, and a Control Data 9427H cartridge disk drive. All the boards in the subsystem are Multibus compatible. A block diagram of the subsystem is presented in figure 6.

The SBC-86/12A microcomputer board is identical to the one used in the communications subsystem hardware section. The serial data channel on the board is connected to the Hazeltine 1520 CRT terminal.

The SBC-534 communications board is also identical to the one used in the communications subsystem. One channel provides for information transfers with the communications subsystem. Another channel is used for communications with the MPS. A third channel is used to output formatted data to the DS-180 printer. The fourth channel is not used at present, but could be used to transfer information with a second communications subsystem.

The model 410 disk controller consists of a microcomputer and drive circuitry dedicated to controlling all the operations (i.e., disk read, disk write, format,

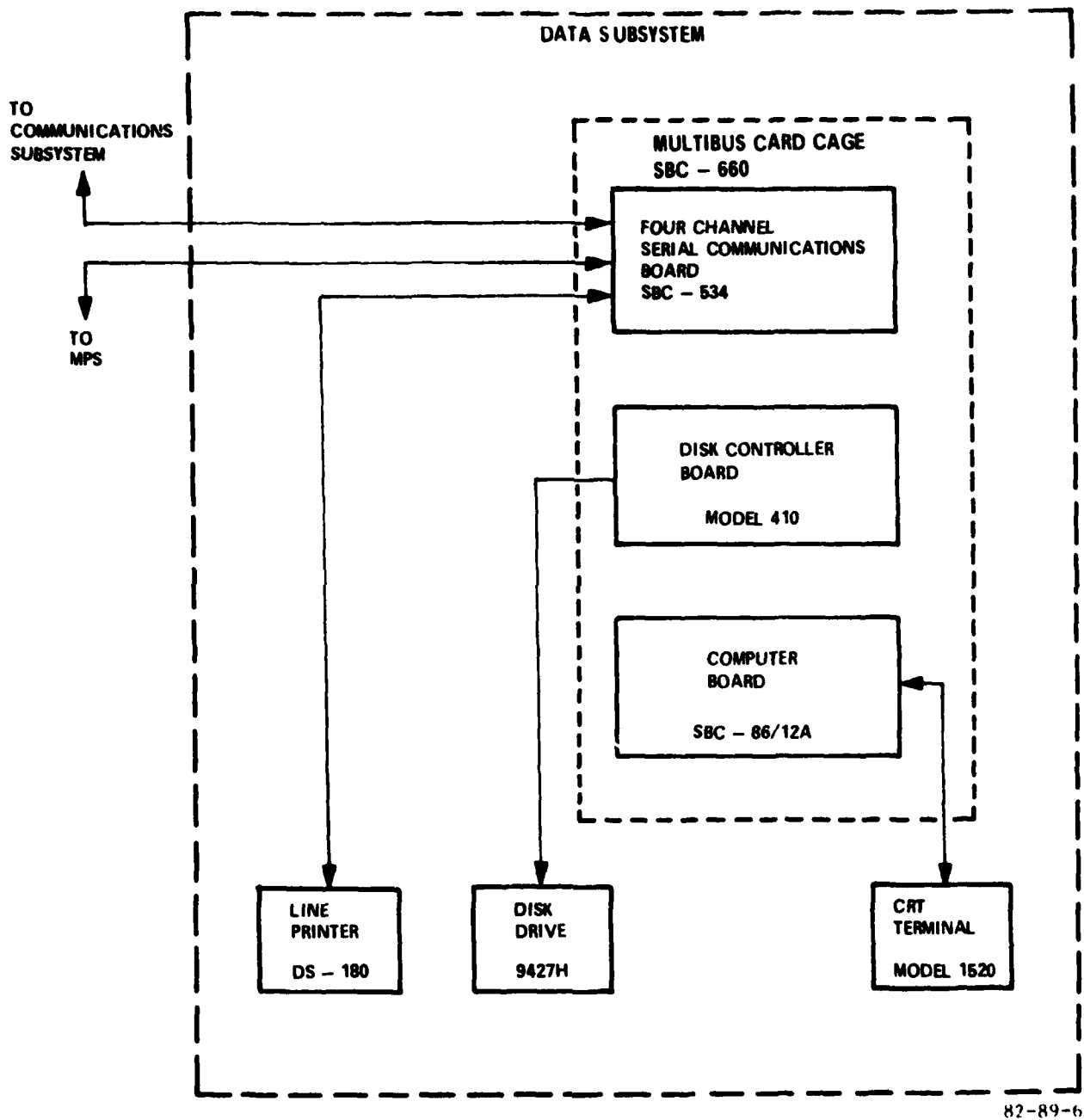


FIGURE 6. DATA SUBSYSTEM BLOCK DIAGRAM

etc.) of the 9427H disk drive. By using direct memory access (DMA) techniques and the Multibus bus, the model 410 controller is able to transfer data to and from RAM memory on the SBC-86/12A computer board and the 9427H disk drive without interfering with the programming operations being performed by the 8086 on the SBC-86/12A board.

The 9427H cartridge disk drive contains 10 megabytes of storage. Five megabytes are on a fixed disk and the remaining five megabytes are on a removable cartridge.

The Hazeltine 1520 is an RS-232C compatible CRT terminal. Any RS-232C compatible terminal may be used. Similarly, the Data South DS-180 is an RS-232C line printer. Any RS-232 compatible line printer may be used.

The SBC-660 chassis is identical to the one used in the communications subsystem. Both subsystems used entirely off-the-shelf boards, chassis, etc. In this way in-house hardware design and assembly was minimal.

SOFTWARE. The data subsystem software consists of an initialization routine, a main routine, a receive interrupt routine, and a real-time clock interrupt routine. All software was written in PLM-86. It consists of approximately 15K bytes of executable code stored in PROM and approximately 17K bytes of data storage in RAM. Approximately two megabytes of storage on the removable cartridge disk is used for long term storage of information from remote site computers.

System operation begins by depressing the reset button on the SBC-660 front panel. The initialization routine is immediately entered. This routine sets all storage areas and program status flags to their appropriate values. The programmable I/O devices on the computer board and on the communications board are initialized. Interrupts are enabled and program execution transfers to the main routine.

A simplified flowchart of the main routine is shown in figure 7; operation is as follows:

1. MOVBLK is called. This routine determines if a complete block of valid data has been received from the communications subsystem (a receive interrupt routine acquires and error checks data from the communications subsystem). If a complete block has not been received, the program execution returns to the main routine. If a block has been received, MOVBLK transfers these data from the receive buffer to one of several central buffers. Central buffers are divided into alarm buffers, polled data buffers, message buffers, etc. By separating buffers according to type, a priority scheme can be developed to allow alarm blocks to receive the fastest processing. After transferring data to the central buffers, program execution returns to the main routine.

2. TRANSMIT is called to respond to any type of request from the communications subsystem or MPS. If a response is already in progress, TRANSMIT will output a character from the appropriate transmit buffer (if the serial channel used is ready to transmit) and return to the main routine. If the transmit buffer is empty, then TRANSMIT determines if a response should be sent and, if so, what type of information. The type of information sent depends on: (a) the type of information just received from the communications subsystem or MPS, (b) if errors occurred in that received data, or (c) if a polled data request or message has been entered at the CRT terminal. After selecting and formatting the appropriate reply, the reply is placed in the appropriate transmit buffer. Program execution returns to the main routine.

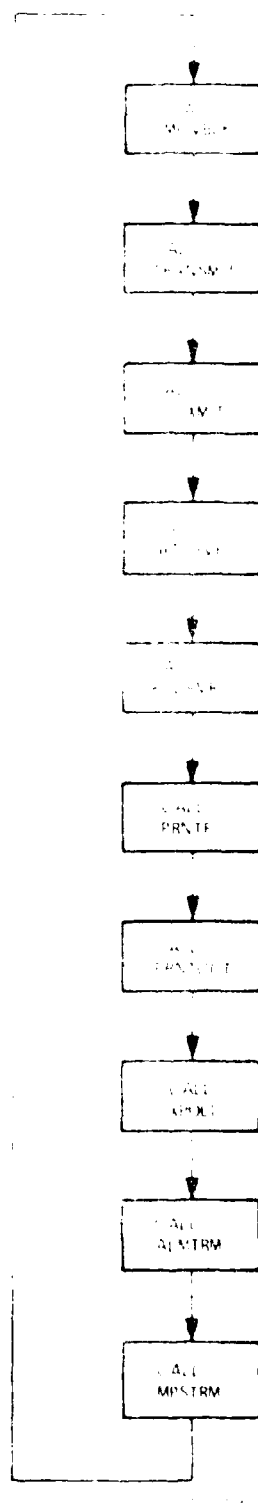


FIGURE 1. DATA SUBSYSTEM MAIN PROGRAM FLOWCHART

3. LOCKMIT is called. If there are data in the CRT terminal transmit buffer and the appropriate serial port is ready to transmit, a character is written to the CRT terminal.

4. After returning to the main routine from LOCKMIT, the routine WTDVR is called. This routine allows data in the central buffers to be transferred to the disk. Upon entering the WTDVR, the disk drive status is tested. If the drive is busy, the program execution returns to the main routine. Otherwise, it performs a series of tasks to determine where on the disk the data should be sent. WTDVR sends the appropriate commands to the disk controller board to enable the controller to write the data from the computer board memory onto the disk or to read directory information from the disk to the computer board memory. Each time a disk operation is initiated, program execution returns to the main routine. A status word, WTOP, keeps track of which disk operation is being performed.

5. After returning from WTDVR, RDDVR is executed. The purpose of this routine is to read a complete set of data from the disk to the computer board RAM. (In some cases, four data blocks may be needed to comprise a single set of data. Data blocks are 256 bytes, except for the last block which may be less.) Once it is determined that a complete set of data is on the disk, RDDVR performs a series of tasks to transfer the data from the disk to a format buffer in RAM. Transfers are made one block at a time until all the blocks of the set have been transferred to the format buffer. Each time a disk operation is initiated, program execution returns to the main routine. A status word, RDOP, keeps track of which block of data is being read.

6. PRNTF is called next. This routine formats the data that has been transferred to the format buffer. PRNTF determines which remote monitor sent the data and the type of data (alarm, polled, or message). Once this is accomplished, the appropriate format routine is called to convert the raw data into data that can be sent to a line printer (i.e., titles are put in place, status bits converted to "ON" or "OFF", etc.). As data are formatted, it is transferred to the print buffer, also in RAM. Once formatting is completed, execution returns to the main routine. Figure 8 shows the difference between raw DME data and DME data formatted by this routine.

7. Next PRNTOUT is called. If there are data in the print buffer and the serial port to the line printer is ready to transmit, a character is sent to the line printer. The program execution then returns to the main routine. This procedure is repeated until the print buffer is emptied. Samples of formatted data outputs and messages are presented in appendix A.

8. After PRNTOUT, XPOLL is executed. This routine examines the central buffer containing communication status information. If a remote monitor serial channel has communications problems or communications has resumed, XPOLL will formulate a message to be displayed on the CRT terminal. The message formulated by XPOLL is placed in the terminal transmit buffer. During execution of LOCKMIT, the message is transmitted to the CRT terminal (see figure 9).

9. ALMTRM is called next. This routine examines the alarm status word for each channel. This status word is set in the MOVBLK routine when the first block of alarm information from a channel is received. If an alarm status word is set, ALMTRM will place an alarm indication message in the terminal transmit buffer. During execution of LOCKMIT, the message is transmitted to the terminal (see figure 10).

10. The last routine called is MPSTRM. If a message from the MPS has been received, this routine places the contents of the message in the terminal transmit routine. During the execution of LOCXMIT, the message is transmitted to the CRT terminal (see figure 11).

MESSAGE FROM MPS:

This is a message from the mps simulator to the concentrator.

82-89-11

FIGURE 11. MESSAGE FROM THE MPS SIMULATOR DISPLAYED ON DATA SUBSYSTEM TERMINAL.

11. When program execution returns from MPSTRM, MOVBLK is called again and the process described in items 1 to 10 is repeated.

During execution of the main routine, three interrupt routines may occur: the receive interrupt, the CRT terminal interrupt, and the real-time clock interrupt. When one of these interrupts occurs, program execution transfers from the main routine to the appropriate interrupt routine.

Each time a character is received from the communications subsystem, the receive interrupt routine is entered. This routine examines the received character to determine if it is a control character (such as a start of text (STX), end of transmission (EOT), or a data character. Error checking in the form of a checksum is performed. Characters forming data blocks are stored in the receive buffer. After the interrupt is serviced, program execution returns to the main routine.

The CRT terminal interrupt routine is entered every time a character is typed on the terminal. This routine examines the character and formulates the appropriate response message. The message is placed in the CRT terminal transmit buffer and transmitted to the terminal when the main routine executes LOCXMIT. Typical messages and responses are shown in figure 12. After servicing the interrupt, program execution returns to the main routine.

The real-time clock interrupt occurs every 50 milliseconds. It contains the software timers required for periodic polls. It also updates the ASCII character real-time clock. This ASCII clock is used to synchronize time with the communications subsystem and all the remote monitors.

REMOTE MONITOR SIMULATORS

To develop and test the concentrator, it was necessary to develop devices that could simulate the communications information provided by remote monitors. Three types of simulators were developed. The first type provided a single channel of simulated very high frequency omnidirectional radio range (VOR)/distance measuring equipment (DME) data. The second type provided four channels of simulated data, one for each of the following: (1) inner marker, (2) outer marker, (3) far field monitor and middle marker, and (4) medium intensity approach lighting system (MALSR). The third type provided 24 channels of simulated data, eight channels each of the following: (1) outer marker, (2) far field monitor and middle marker, and (3) MALSR.

TYPE H FOR HELP

H

TYPE ONE OF THE FOLLOWING:

U-TO POLL A REMOTE SITE

P-TO CREATE A PERIODIC POLL

S-TO SEND A MESSAGE TO A REMOTE SITE

T-TO ENTER TIME

CONTROL/D TO ABORT ANY OPERATION

U ENTER ADDRESS- 20 OK

P ENTER ADDRESS- 5

ENTER POLLING INTERVAL IN HOURS- 8 OK

S

ENTER THE DESTINATION ADDRESS 7 ENTER MESSAGE - TYPE CONTROL/C TO TERMINATE

This is a message from the concentrator to remote site #7.

OK

T

ENTER TIME- 00:30:00 OK

82-89 12

FIGURE 12. DATA SUBSYSTEM LOCAL TERMINAL COMMANDS

This section will describe the tasks performed by the simulators and discuss the hardware and software development of each type of simulator.

REMOTE MONITOR SIMULATOR TASKS.

1. Communicate with the concentrator.
2. Error check information received from the concentrator.
3. Upon receiving a request from the concentrator, transmit periodic information to the concentrator.
4. Upon receiving an alarm command from the local terminal keyboard, send simulated alarm information to the concentrator.
5. Receive messages entered at the local terminal keyboard and send these messages to the concentrator.
6. Receive messages from the concentrator and display the contents of the message on the local terminal.
7. Upon receiving the proper keyboard entry at the local terminal, send to the concentrator a request for the data from another remote monitor (this capability does not exist on the one-channel simulator).
8. Provide a real-time ASCII character clock. The clock is sent to the concentrator with alarm or polled information. Since the clock changes each second, some of the information sent to the concentrator is "live."

HARDWARE DESIGN.

ONE-CHANNEL AND FOUR-CHANNEL SIMULATORS. The hardware designs for both the one-channel and four-channel simulators are very similar. Both devices employ the Zilog Z-80 microprocessor. Each simulator consists of three boards manufactured by Zilog: a microcomputer board (MCB), a serial interface board (SIB), and a program-mable read only memory board (PMB). A Zilog SCC-9 card cage was wired to accommodate the three boards. In addition, a Texas Instruments Silent 700 portable terminal was used as a local terminal.

The MCB consists of the eight-bit Z-80 microprocessor, 4K bytes of RAM, and sockets for 4K bytes of PROM. In addition, the board contains a system clock, a counter/timer circuit, a serial data channel, and four switches. The counter/timer circuit was used to create real-time clock interrupts and also to generate a baud-rate clock for the serial data port. The serial data channel communicates with the local terminal.

The SIB consists of four serial data channels, a system clock, a counter/timer circuit to provide baud rates to each serial channel, and another counter/timer circuit to generate interrupts when data are received by the serial data channels. The serial channels on this board are used to communicate with the concentrator.

The one-channel simulator uses only the first serial data port on the SIB, the four-channel simulator uses all four serial ports. For the four-channel simulator, the switches on the MCB are used to determine whether a 300 or 1200 bps rate will be used to communicate with the concentrator.

The PMB provides sockets to expand the system program memory (PROM) by 32K bytes.

The SCC-9 card cage contains nine 122-pin card slots. The backplane contains 5 volt and ground busses. All other backplane connections had to be wire wrapped. Power requirements to the card cage include +5 volts, +12 volts, -5 volts, and -12 volts.

The local terminal generally used was a Silent 700. Any RS-232C compatible terminal can be used. For some tests, terminals from Computer Devices Incorporated (CDI) and Applied Digital Data Systems (ADDS) were used.

The computer hardware described above was selected because it was immediately available (the parts were spares from earlier RMMS activities). Thus, it kept project costs down.

A block diagram of the simulator hardware is shown in figure 13. A pair of both types of simulators were built. A photograph of typical simulators is presented in appendix B, figure B-2.

TWENTY-FOUR CHANNEL SIMULATOR. Unlike the one-channel and four-channel simulators, the twenty-four channel simulator uses the 8086 microprocessor. This device uses the same type of Multibus boards as the concentrator. It includes the SBC-86/12A microcomputer board and three BI018 eight-channel serial communications boards. The boards reside in an Intel ICS-80 Multibus chassis. A portable CRT terminal from ADDS is also used, however, any other RS-232C terminal can be used. A block diagram of this simulator is shown in figure 14.

SIMULATOR SOFTWARE.

SINGLE-CHANNEL SIMULATOR. The software consists of an initialization routine, a main routine, a receive interrupt routine, and a real-time clock interrupt routine. The software was written in the Z-80 assembly language and in PLZ, a high level language for the Z-80. The executable code requires over 7K bytes of PROM. Data storage uses 2.6K bytes of RAM.

The initialization routine programs the MCB counter/timer circuit to cause an interrupt every 20 milliseconds. The interrupt generated will be used to trigger the real-time clock interrupt routine. The initialization routine then programs the MCB serial data channel and another channel of the counter/timer circuit for 300 bps asynchronous communication with the local terminal. The first channel on the SIB and the appropriate SIB counter/timer circuit is programmed to provide 300 bps asynchronous communication with the concentrator. A second counter/timer circuit on the SIB is programmed to generate an interrupt every time the serial channel receives a character from the concentrator. The initialization routine then sets software status flags to be used in the other routines, enables interrupts, and transfers control to the main routine.

A simplified flow chart of the main routine is shown in figure 15. Operation is as follows:

1. The routine CRTRCV is called. This routine acquires data from the local terminal via the serial channel on the MCB. CRTRCV formulates a response to a local terminal request and provides the address location of the response. This routine also sets a status word to let an output routine to the concentrator know

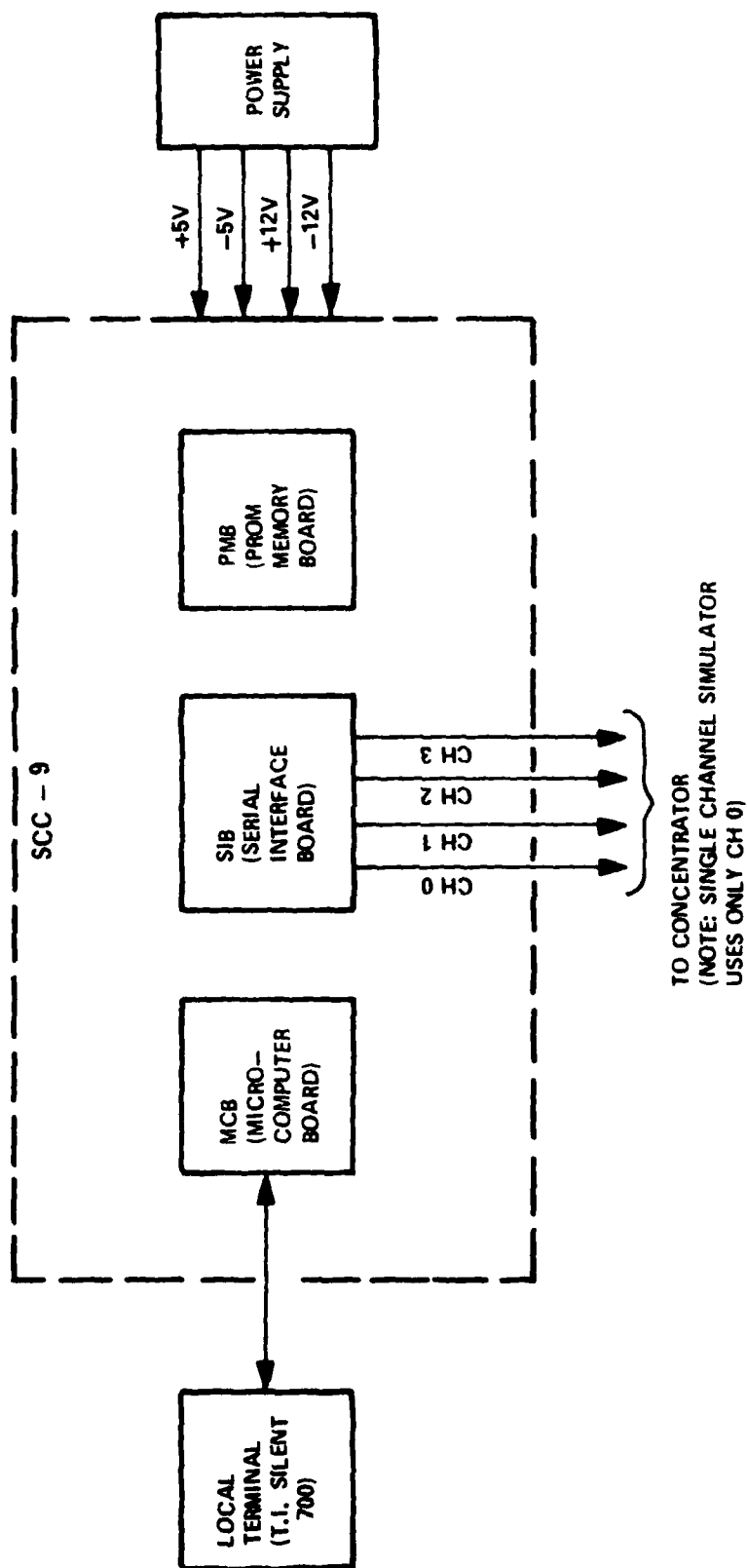
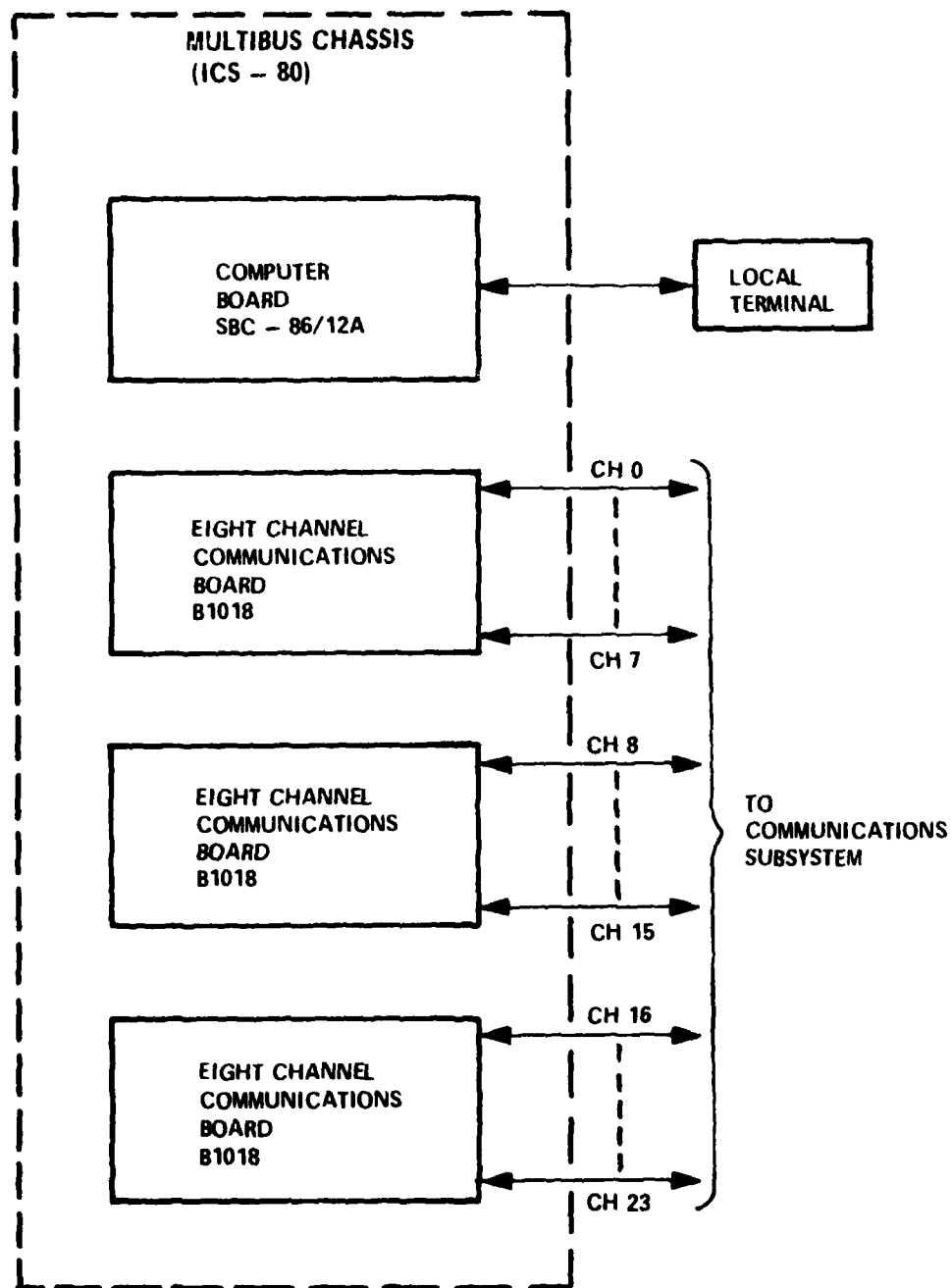


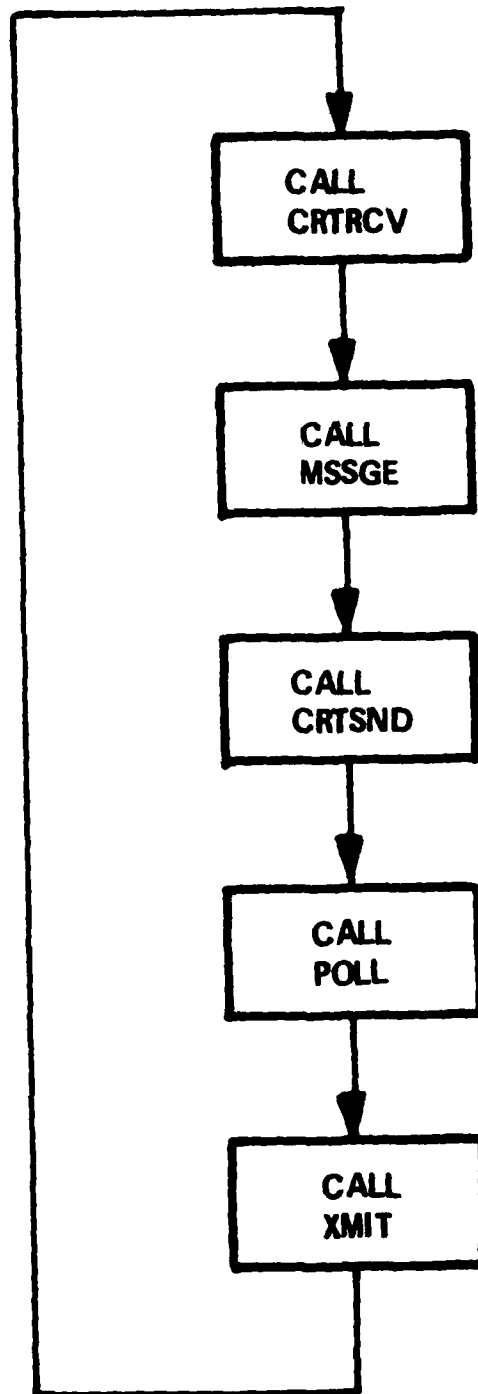
FIGURE 13. Z-80 REMOTE MONITOR SIMULATOR BLOCK DIAGRAM

82-89-13



82-89-14

FIGURE 14. TWENTY-FOUR CHANNEL REMOTE MONITOR SIMULATOR BLOCK DIAGRAM



82-89-15

FIGURE 15. SINGLE-CHANNEL SIMULATOR MAIN ROUTINE FLOWCHART

11. In addition, the message is pending. Attached is the VLS. I will let you know if I receive any more information.

the terminal of the other cell (i.e., $M_{\text{source}} = M_{\text{target}}$). In this case, the data_id that originated at another simulator is the same as that of the local terminal, so the transfer request to the local terminal will be rejected. In the case of a transfer request, sent by the receiver interrupt routine, M_{source} will be the data_id of data at the terminal of another terminal. In this case, the data_id is not the same as that of the data in the local terminal, so the transfer request will be accepted. In that case, the term exchange_return is transferred to the terminal of the other cell in the buffer. After the exchange term is transferred, the transfer is being.

[illegible]

As a result, the AMU Series information in the concentration of the GRSN's funds, in relation to the four terms, 11, 10, 9, and 8, is better and the overall means in the six is ready to transmit, and for the concentration. After sending the information of 11 the Series is ready to transmit, in an executive policy, to the main holding.

The main routine continually executes the routine `wait_for_interrupts`. In this operation, interrupts may occur that temporarily suspend the execution of the current routine. Every time a signal is received from the hardware, the routine `interrupt_routine` is activated, and every time the signal is cleared, the interrupt is deactivated.

The receive interrupt routine places all incoming data in a buffer and identifies the type of information received. It then determines if a general poll, a request for information, or a request for acknowledgment (negative or knowledge) or a test status (data received or simulator) has been received from the concentrator. The interrupt status is reset so that the main routine will be able to use the received data properly. If a message was received, the receive interrupt routine transfers contents of the message to the local terminal transmit buffer. After the receive interrupt routine is completed, program execution returns to the point in the routine where it had been when the interrupt occurred.

The counter/timer circuit on the MCB triggers an interrupt every 10 milliseconds, placing program execution in the real-time clock interrupt routine. This routine performs several tasks. Each time the routine is entered, a software timer incremented by one. When the timer reaches 50 (which happens once a second), the timer is reset to zero and an ASCII character real-time clock is updated.

If an alarm is to be simulated, the contents of the ASCII clock buffer are also moved to another buffer. This time value will be sent with simulated alarm data to the concentrator. During the CRTRCV routine, a time value may be entered from the local terminal. This routine replaces the current ASCII clock value with the value obtained from the local terminal.

FOUR CHANNEL SIMULATOR. The basic procedures followed in the single-channel simulator are also used in the four-channel simulator. Most routines were modified to handle four simulated channels instead of only one. As a result, more executable code and additional storage buffers were required. The four-channel simulator uses over 8.5K bytes of PROM and almost 4K bytes of RAM.

In addition to increasing the buffer storage and modifying the routines to handle four channels, there are several other differences in the four-channel simulator: (1) during the initialization routine, switch settings on the MCB are checked and used to program the baud rate clocks to either 1200 or 300 bps on the four channels; (2) each channel contains its own set of simulated data; (3) information sent to the concentrator always includes the date and time; (4) by making the proper keyboard entries, all four channels can be made to alarm simultaneously; and (5) using the local terminal, information from other remote simulators can be requested by any of the four-simulator channels.

TWENTY-FOUR CHANNEL SIMULATOR. Although this simulator uses different hardware, a different language (PLM-86), and has many more channels than the Z-80 based simulators, the principles of operation and software flowcharting are nearly identical to the Z-80 based simulators. The software for this simulator requires approximately 6K bytes of PROM and 14K bytes of RAM.

The only difference between this simulator and the four-channel simulator are: (1) the number of channels in the simulator, (2) the data rates can be set to 110, 300, or 1200 bps through terminal keyboard entries, and (3) the terminal is serviced via an interrupt routine instead of a CALL statement in the main routine.

All 24 channels can be made to alarm simultaneously by typing an "A" (to signify an alarm request) and then typing a "Z."

Figure 16 summarizes the types of terminal operations that can be performed on the 24-channel simulator local terminal. Terminal operations for the Z-80 based simulators are very similar.

MAINTENANCE PROCESSOR SUBSYSTEM SIMULATOR

The MPS simulator was developed to demonstrate and test the data subsystem's ability to respond to a higher level processor's requests. No effort was made to make the simulator act like an MPS. It is simply a communications device designed to acquire information from and exchange messages with the data subsystem.

The simulator performs the following tasks:

1. Communicates with the data subsystem.
2. Error checks information received from the data subsystem.

H
TYPE ONE OF THE FOLLOWING:
A-TO CREATE AN ALARM
B-TO REQUEST DATA
D-TO ENTER THE DATE
R-TO CHANGE BAUD RATES
S-TO SEND A MESSAGE
T-TO ENTER TIME

A WHICH CHANNEL? A OK

B WHICH CHANNEL? C
ENTER TWO DIGIT DESTINATION ADDRESS- 50 OK

FAC IDENT- 5

TIME: 08:59:12

DATE: 03/25/82

INNER MARKER

LATEST

/	MAIN	/	STDBY	/	OFF	/	ABN MON/	ABN FE/	PERF	/
	ON		OFF		OFF		OFF	OFF	OFF	

D ENTER DATE- 03/25/82 OK

R WHICH CHANNEL? E TYPE 1 FOR 110, 2 FOR 300, 3 FOR 1200 2 OK

S WHICH CHANNEL? F
ENTER DESTINATION ADDRESS- 4
ENTER MESSAGE- TYPE CONTROL/C WHEN FINISHED
This is a message from remote site F to remote site 4.
OK

T ENTER TIME- 09:00:00 OK

82-89-16

FIGURE 16. REMOTE MONITOR SIMULATOR LOCAL TERMINAL COMMAND

3. Receives messages entered at the MPS's local terminal and sends these messages to the data subsystem.

4. Receives remote monitor information and messages from the data subsystem.

Although the MPS simulator checks all incoming information, the simulator was not programmed to display this information.

The MPS simulator hardware block diagram is shown in figure 17. It consists of a Z-80 MCB computer board and an SIB serial communications board that reside in an SCC-9 card cage. These components are identical to those used in the Z-80 remote monitor simulators. In addition, an RS-232C compatible terminal is required.

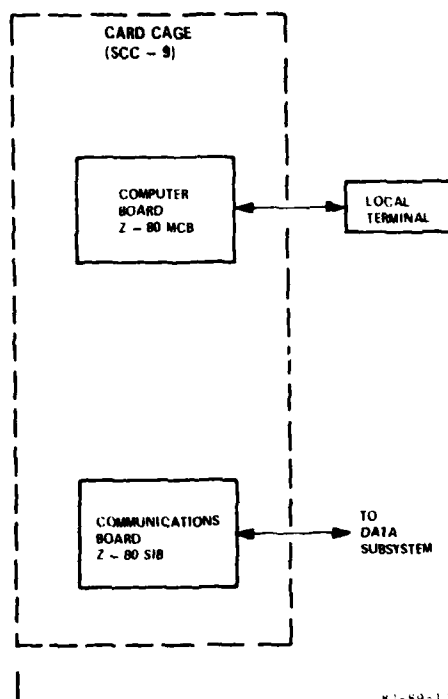
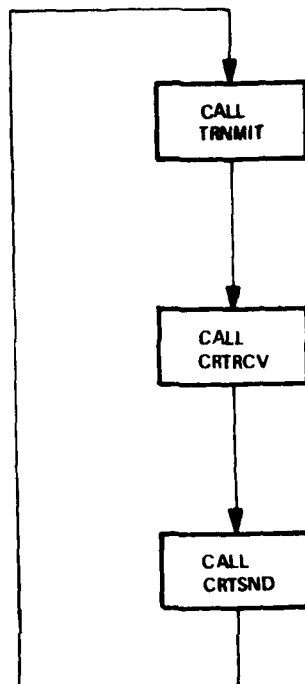


FIGURE 17. MPS SIMULATOR BLOCK DIAGRAM

The MPS simulator software is written in the Z-80 assembly language and in PLZ. It requires less than 4K bytes of PROM and about 1K bytes of RAM. The program consists of initialization, main, receive interrupt, and real-time clock routines.

The initialization routine programs the counter/timer circuit on the MCB to generate interrupts every 20 milliseconds. It also initializes both the serial channel on the MCB (which services the local terminal) and a channel on the SIB (which communicates with the data subsystem) for a data rate of 300 bps. Storage areas are cleared, status flags are set to their initial values, and then program execution is transferred to the main routine.

The main routine consists of only three subroutines, as shown in the simplified flowchart (figure 18). The TRNMIT routine determines the type of information sent to the data subsystem and also controls the transmission of this information through the SIB serial channel.



82-89-18

FIGURE 18. MPS SIMULATOR MAIN ROUTINE FLOWCHART

The CRTRCV subroutine inputs characters from the local terminal through the MCB serial channel. It formulates the appropriate response to the keyboard entries and places this response in the terminal transmit buffer.

The response to terminal keyboard entries are controlled by CRTSND. If the MCB serial channel is ready to transmit, CRTSND sends a character from the terminal transmit buffer through the MCB channel to the terminal.

The three subroutines described are continuously repeated. Execution of these routines may be temporarily disabled by the occurrence of a receive interrupt from the data subsystem or an interrupt from the MCB counter/timer circuit.

Each time a character is received from the data subsystem, the receive interrupt routine is entered. This routine error checks the incoming characters and provides status information to the TRNMIT subroutine. TRNMIT uses the status information in formulating a response.

The real-time clock interrupt is entered every 20 microseconds. This routine updates a timer used in generating polls to the data subsystem.

CONCENTRATOR TESTING

The concentrator testing consisted of: (1) verifying that the concentrator performed as designed, and (2) to determine if it could handle a severe processing load. Much of the concentrator testing was conducted using the configuration shown in figure 19.

The block diagram shows a Hewlett-Packard (HP) model 1640A Serial Data Analyzer inserted in the serial line between the data subsystem and MPS simulator. The HP-1640A displays all the data exchanged between the subsystem and simulator. This device was extremely useful in testing and debugging all system communications software.

Each simulated remote monitor channel is identified by an ASCII character beginning with "1" (31 hexadecimal) and continuing through "S" (53 hexadecimal). This identification is transmitted as part of the header when information is transferred to the communications subsystem. The data subsystem is identified as channel "0" (30 hexadecimal). Since only 34 simulator channels were available, one channel (channel "3") does not service a remote monitor. However, during the tests that will be described, all channels, including "3", were checked.

Using the configuration shown in figure 19, by inserting the HP-1640 in the appropriate serial data line and making entries on the appropriate terminals (see figures 3, 12, and 16), the following were demonstrated:

1. In the absence of any alarms, messages, etc., all remote monitors received general polls at the selected interval (from 1 to 9 seconds).
2. The general poll rate interval could be varied by making entries on the communications subsystem terminal.
3. Periodic polls were automatically performed by the communications subsystem at the selected polling interval (from 1 to 99 hours).
4. Periodic polling intervals could be varied by keyboard entries on the communications subsystem terminal.
5. Periodic polling intervals could also be varied by keyboard entries on the data subsystem terminal.
6. All polled and alarm information were sent to the data subsystem via the communications subsystem.
7. All polled and alarm information received by the data subsystem were stored on the disk, formatted and displayed on a printer, and retransmitted to the MPS simulator.
8. Messages from remote monitors were sent via the communications subsystem to the proper destination (another remote site or the data subsystem).
9. Messages from the data subsystem could be sent to any remote monitor or the MPS simulator.

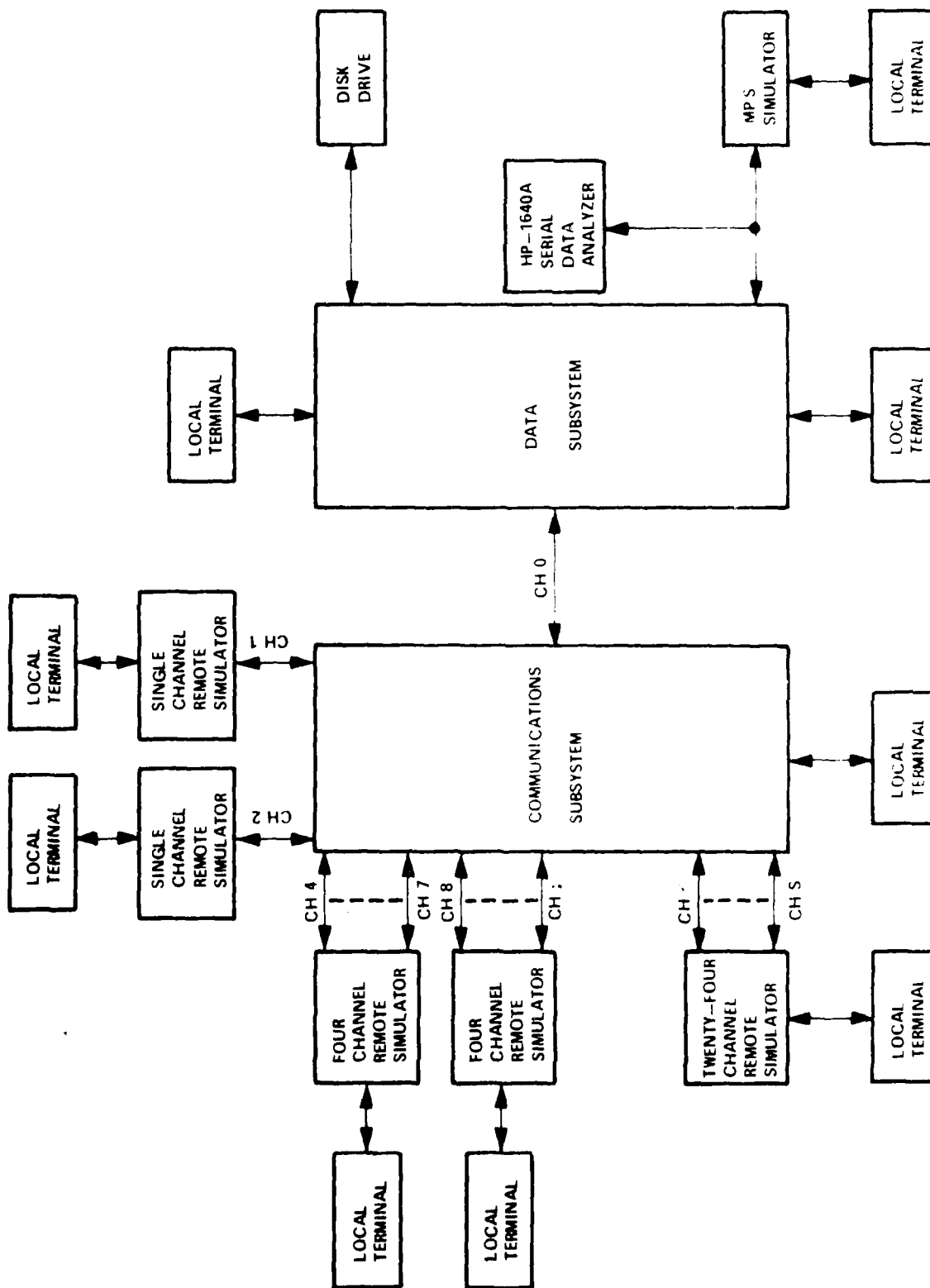


FIGURE 1.4. SIMULATION TEST CONFIGURATION

10. Messages from the MPS simulator could be received by the data subsystem.
11. Through terminal entries at one remote monitor, the data at a second remote monitor can be requested. The communications subsystem sends this request to the data subsystem. The data subsystem initiates a periodic poll request for the second monitor. The data returned are formatted and transmitted back (through the communications subsystem) to the requesting remote monitor.
12. Upon receiving alarm and polled information and messages (addressed to the data subsystem) simultaneously, the communications subsystem sent alarm information to the data subsystem before sending the polled information or message. Polled information was sent before the message.
13. The communications subsystem can simultaneously report alarms to the data subsystem and route messages from one remote monitor to another remote monitor.
14. All devices can recover from communications errors caused by temporarily disconnecting serial data lines.
15. The data rates on all eight-channel serial communications boards (in both the communications subsystem and the 24-channel remote monitor) could be changed by making keyboard entries.
16. The number of remote monitor channels polled by the communications subsystem can be changed by making keyboard entries at the communications subsystem terminal.
17. An appropriate message is displayed in the data subsystem terminal when communications to a remote monitor simulator is lost or resumed.
18. An alarm message is displayed on the data subsystem terminal after the data subsystem receives the first block of alarm information.
19. Date and time may be entered at remote monitor terminal keyboards or at the communications subsystem terminal keyboard.
20. Time may be entered at the data subsystem terminal keyboard. During periodic polls, data subsystem time is sent to the communications subsystem and remote monitors in order to synchronize all real-time clocks.
21. Any remote monitor may be polled for information by making the appropriate keyboard entries on the data subsystem terminal.

In addition to these tests, the concentrator was run continuously for over 50 hours. This verified the reliability of the system hardware.

The above tests indicated that the concentrator was performing as designed. The next test was to determine the concentrator's ability to handle unusually heavy communications loads.

To do this, the data rates on 32 remote channels were set to 1200 bps. The remaining two channels were set to 300 bps (the one-channel simulators can only operate at 300 bps). The general poll rate was set for one poll per second.

12
Alarms were then initiated on all channels simultaneously. The information received at the data subsystem and printer indicated that the concentrator successfully processed this unusual load.

According to a study conducted as part of the concentrator subtask (see reference 3), it is unlikely that a concentrator in the field will require as many as 34 channels. Also, the data rates will probably not be as high as those used in this test. It is also unlikely that a condition will ever exist in which 34 channels are transmitting alarm data simultaneously. For these reasons, it can be concluded that this concentrator design is fast enough to handle any task that is required.

ANALYSIS OF CONCENTRATOR

SYSTEM FLEXIBILITY.

In the "Introduction" to this report, it was stated that the concentrator designed at the Technical Center should perform three functions:

1. Communicate with remote monitors and an MPS.
2. Store the information obtained from remote monitors.
3. Format and display the collected information.

Since applications exist where all three functions are not required, the concentrator design had to be flexible, modular, and expandable. In order to meet these requirements, the Multibus bus structure was chosen.

The discussion in this section will show that by taking advantage of the Multibus capability and of the modular subsystem design employed in the Technical Center's concentrator design, the concentrator can be reconfigured to handle several other potential applications.

Four possible configurations of the concentrator will be discussed:

1. Basic concentrator configuration.
2. Communications and data subsystems in a shared chassis.
3. Communications subsystem only.
4. Communications and display system.

A summary of these configuration options and potential applications are presented in table 2. A breakdown of component costs for the Technical Center's concentrator is shown in table 3.

BASIC CONCENTRATOR CONFIGURATION. This configuration is the one developed at the Technical Center and already described in detail in previous sections.

It should be noted that the communications and data subsystems are housed in separate chassis. It is not even necessary that they be collocated. In this way, a communications subsystem could be located at a small airport or some other remote facility, while the data subsystem is located at a large facility. The communications subsystem could then be used to consolidate data links. Many remote site

TABLE 2. SUMMARY OF CONCENTRATOR OPTIONS

<u>Option</u>	<u>Approximate Cost</u>	<u>Potential Applications</u>
1. Basic Configuration	20K	Large airports with many remote sites
2. Communications and Data Subsystems Share Same Chassis	18K	Medium to large airports with remote sites requiring fast processing
3. Communications Subsystem-Only	5K to 7K*	Small airports and remote facilities*
4. Communications and Display	6.5K to 10K**	Medium size airports

*Depends on number of channels used.

**Depends on number of channels used, on memory size, and number of computer boards.

TABLE 3. COST BREAKDOWN FOR THE TECHNICAL CENTER'S CONCENTRATOR

<u>Communications Subsystem</u>	<u>Cost</u>	<u>Data Subsystem</u>	<u>Cost</u>
Computer Board	2.5K	Computer Board	2.5K
Communications Boards	2.5K*	Communications Board	0.75K
Multibus Chassis and Power Supply	2K	Disk Controller Board	2K
Subsystem Total	\$7K	CRT Terminal	1K
		Printer	1K
		Ten Megabyte Disk	4.25K
		Multibus Chassis and Power Supply	2K
		Subsystem Total	\$13.5K

Concentrator Total \$20.5K

*Cost required to support a 35 channel system

channels at the small airport would be sent to the communications subsystem. This subsystem can prioritize the information and send it over a single long distance link to the data subsystem.

Another advantage of this configuration is that the data subsystem can receive inputs from more than one communications subsystem.

A potential application for this concentrator would be at large airports or other manned facilities. In this application, there will probably be a large number of remote monitor channels and a need to display and store the collected information locally.

COMMUNICATIONS AND DATA SUBSYSTEMS IN A SHARED CHASSIS. By taking advantage of the Multibus, both the communications and the data subsystems can be placed in the same chassis (as shown in figure 20). In doing so, the cost of the second chassis (about \$2,000) is eliminated. Data can be exchanged between subsystems by either using the same serial technique used in the basic configuration, or by having both processors share common memory locations. In the latter approach the communications subsystem computer would store data in a common memory area. The data subsystem computer would read the data from the common memory and perform the storage and display/formatting functions. This approach provides the fastest processing of the data, since data received by the communications subsystem is immediately available to the data subsystem.

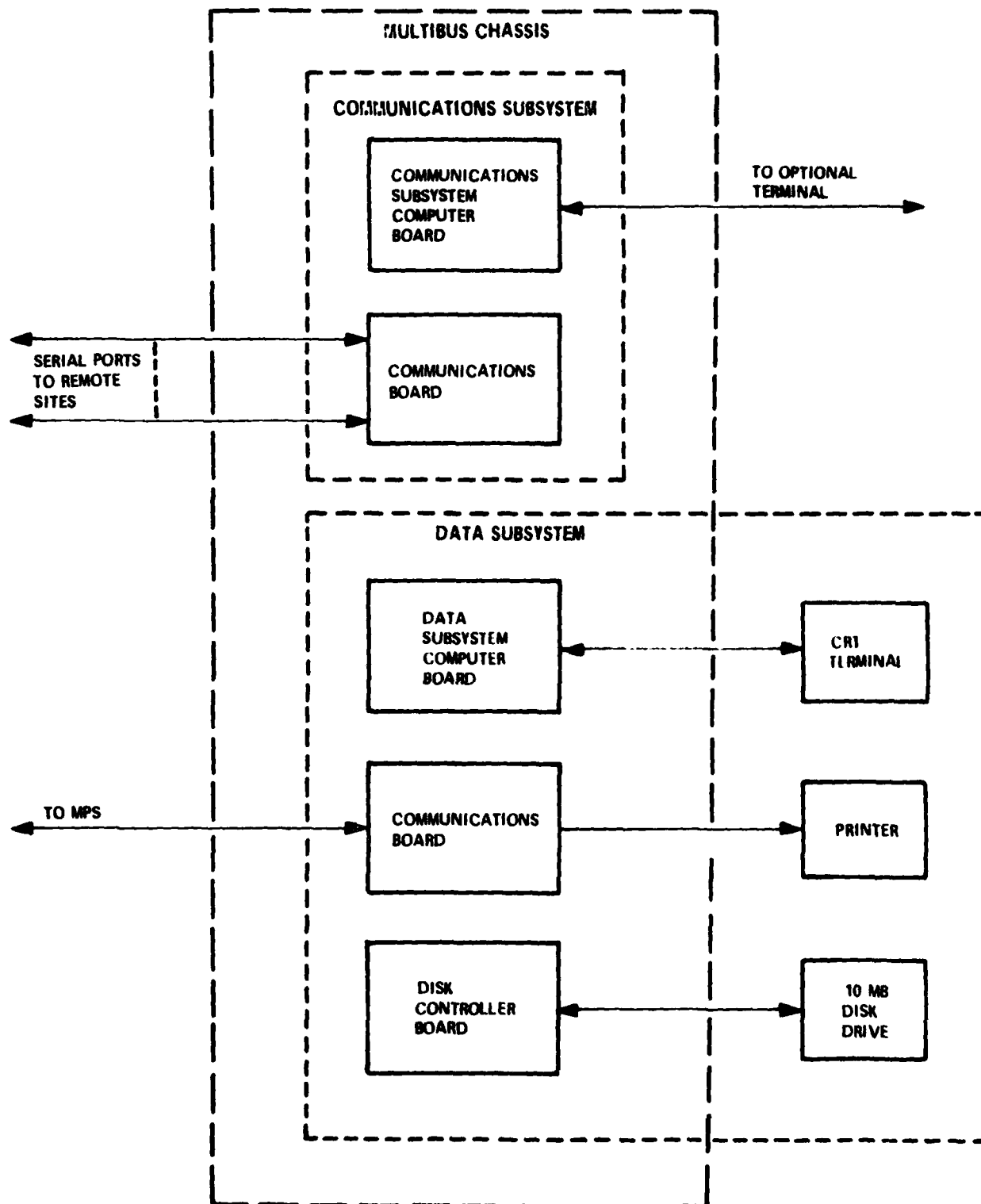
Applications for this option would be similar to those discussed in the first option. This system would be more favorable where the fast transfer of information is desirable. It should also be used instead of the first option, when only a limited number of remote facilities are to be serviced. Since both subsystems occupy the same chassis, communications channel expansion is limited.

COMMUNICATIONS SUBSYSTEM ONLY. For some applications, it may not be necessary for the concentrator to store and display information. For these communications-only applications, the data subsystem can be eliminated. Information collected from remote monitors by the communications subsystem is sent directly to the MPS. By eliminating the data subsystem, the cost of the concentrator is substantially reduced.

One potential application for this subsystem is as an intelligent front-end communications processor for the MPS. If it is necessary for the MPS to service a large number of remote monitor channels, the communication subsystem could be used to consolidate the channels, thus, relieving the MPS of much of the communications overhead.

A similar application exists at remote sites or small airports where data link costs would be reduced by consolidating several remote monitor channels into one output channel.

COMMUNICATIONS AND DISPLAY SYSTEM. The communications subsystem-only option can be modified to provide both communications and data display. The data display function can be accomplished by adding additional memory, and by adding either data format routines to the existing software or a second computer board with the specialized software. However, adding the extra memory and a second board may limit the number of channels that can be serviced by the device.



82-59-10

FIGURE 20. COMMUNICATIONS AND DATA SUBSYSTEMS SHOWING MULTIBUS CHAINS

The need for the second computer board (as shown in figure 21) would be determined by the number of channels being handled by the communications board. The cost of additional memory is between \$1,500 to \$3,000, depending on the amount needed (up to 0.5 megabytes). The additional computer board is about \$2,100. A terminal would add \$1,000 to \$2,000 to the cost. A portable terminal could be used. If this location is not manned, the terminal can be removed.

A potential application for this configuration would be at medium sized airports that are usually manned. In these applications it may be necessary or convenient to be able to display data locally, but unnecessary to provide long term storage.

ADDITIONAL STORAGE. The last two options discussed do not provide a means for long-term storage of collected data. If it becomes necessary to store data, but it does not have to be accessed at high speed, various tape drives can easily be interfaced to the concentrator. This can be accomplished either by using a serial port or by adding a Multibus controller. In some cases, over a megabyte of data storage can be provided for \$300 to \$400.

Bubble memory boards are also available in Multibus configurations, however, tapes or disks are much more cost effective. In addition, relatively low cost floppy disk drives are available. They may be useful for program storage, but are not generally useful for this type of data storage.

SOFTWARE ANALYSIS.

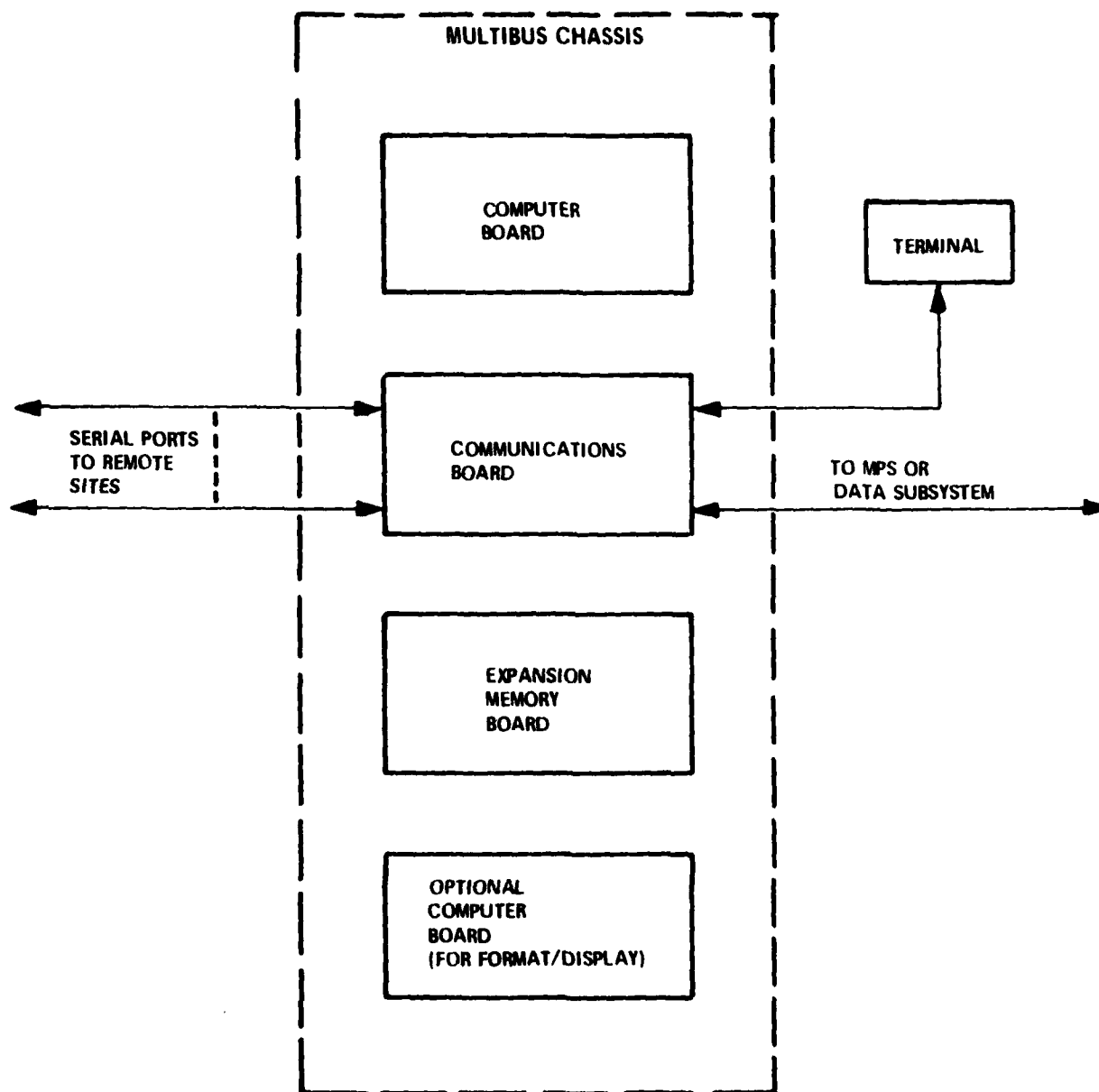
The program for the concentrator was written entirely in PLM-86, a high level language. This language has some advantages and some disadvantages. One advantage is that the instruction set contains statements to handle I/O operations and bit manipulations. PLZ, which was used in the Z-80 simulators, does not have these capabilities.

PLM-86 is structured in a way that necessitates the use of CALL statements instead of GO TO statements. As a result, the programming tends to become more modular than assembly language or FORTRAN. This makes debugging easier. It also makes it easier to add routines to an existing program.

One disadvantage of PLM-86 is that it can only be used in 8086 or 8088 micro-processor-based systems. For field implementation, a programming language such as PASCAL (which is structured like PLM-86) is more desirable. PASCAL compilers exist for most types of computers. Therefore, the concentrator software could be made almost independent of the type of concentrator computer used.

In the data subsystem software, several tasks may simultaneously compete for the use of the subsystem peripherals (i.e., disk drive, terminal, and printer). For example, one section of the program may need to read data from the disk in order to eventually send formatted data to the printer. At the same time, another section of the program may want to write some data just received from the communications subsystem to the disk. In order to allow all sections of the program access to the peripherals, numerous software status words were required to keep track of what operations were being performed on the peripherals.

Using the software approach taken, the programming becomes much more complex as additional competing tasks are added. This problem can be alleviated by adding real-time operating system software to the subsystem. Real-time operating systems



82-89-21

FIGURE 21. COMMUNICATIONS AND FORMAT/DISPLAY SYSTEM

for this application did not exist when the concentrator task was begun, however, they are available now.

In order to use a real-time operating system most effectively, several hardware changes should be made to the data subsystem. The cartridge disk drive should be replaced by a floppy disk drive and a Winchester disk drive. The floppy disk drive is necessary because most microcomputer operating system software are available only on floppy disks. The Winchester drive would be used to replace the cartridge disk as a bulk storage device. A Winchester drive was not a practical alternative when the concentrator task was begun; however, the drop in cost and the development of the 5 1/4-inch Winchester disk has made this approach better than the use of the cartridge disk.

By obtaining the operating system software and disk drives, the data subsystem software can readily be programmed to handle additional tasks. Furthermore, other options would also become available. A compiler and monitor program could be resident in the data subsystem. In this way, software could be written and debugged in the data subsystem. This eliminates the need for a microcomputer development system for data subsystem software development.

With the changes and additions described above, the data subsystem would contain all the hardware and software needed to perform as a very low cost general purpose minicomputer. Such a device could also be used to handle other tasks in addition to on-line remote monitoring (e.g., trend analysis, schedule assignments, etc.).

CONCLUSIONS

The Technical Center's concentrator was developed as a feasibility model. Based on the design, development, and test effort conducted for this subtask, a number of conclusions can be reached.

1. The concentrator approach to Remote Maintenance Monitoring System (RMMS) data communications and processing is feasible and can be used to handle several types of tasks that may be required in this system.
2. A concentrator can be developed using off-the-shelf hardware requiring only minimal mechanical and electrical assembly.
3. The system designed performs the three general functions required:
 - a. Communicate with remote monitors and a maintenance processor subsystem (MPS).
 - b. Store the information obtained from the remote monitors.
 - c. Format and display the collected information.
4. Tests have shown the concentrator can communicate with over 30 remote monitor channels at data rates up to 1200 bits per second (bps). Since this represents a worse case situation, it is concluded that the concentrator can support the requirements of any planned field configuration.

5. The Multibus™ design will allow the system hardware to easily be reconfigured to handle fewer than the three general functions just described. The reconfigured hardware will cost less than the basic concentrator configuration developed at the Technical Center.

6. Any type of microprocessor that is available on a Multibus compatible board will be hardware compatible with the concentrator (provided the processor is fast enough). Similarly, all Multibus compatible input/output (I/O) boards are also hardware compatible. Although the PLM-86 language can only be used with the 8086 and 8088 microprocessors, other languages, such as PASCAL, have compiler programs available for many microprocessors. Therefore, using Multibus hardware and a high-level language such as PASCAL, a concentrator can be designed that will be almost entirely hardware and software independent of the type of microprocessor used.

7. A real-time operating system will simplify program design and facilitate program changes and expansions.

8. The cartridge disk drive can be replaced by a floppy disk drive (as a removable medium) and a Winchester disk drive (for bulk storage). Winchester drive technology has improved rapidly since this subtask began. The cost of floppy and Winchester drives have been dropping and, together, they should cost less than the cartridge disk drive.

9. By using a real-time operating system and the floppy and Winchester drives, the data subsystem can be made to perform as a general purpose minicomputer. It can then be used to handle additional tasks.

10. At the time this subtask was started, ICD-1 was being rewritten. Therefore, the communications protocol used in the concentrator is not entirely ICD compatible.

RECOMMENDATIONS

It is recommended that the following enhancements be made to the concentrator developed by the Technical Center:

1. Develop a concentrator-to-maintenance processor subsystem (MPS) communications interface.
2. Rewrite the concentrator software to be fully compatible with ICD-1.
3. Add autodial/autoanswer software to the concentrator.
4. Replace the cartridge disk drive with a floppy disk drive and a Winchester drive.
5. Add a real-time operating system and a PASCAL compiler to the data subsystem.

REFERENCES

1. Interface Control Document for the Remote Maintenance Monitoring System (ICD-I), National Airspace System Configuration Management Document, NAS-MD-790, April 17, 1981.
2. Level II Interface Control Document Guidelines (ICD-II), National Airspace System Configuration Management Document, NAS-MD-791, September 1978.
3. Wainland, David, Remote Maintenance Monitoring System Concentrator - Data Network Study, FAA Technical Center Letter Report, CT-81-100-9-LR, July 1981.

APPENDIX A

FORMATTED DATA INFORMATION

Figure A-1 displays the information from a combined very high frequency omnidirectional radio range (VOR)/distance measuring equipment (DME) site after the data has been formatted by the concentrator. The VOR information consists of monitor and control panel status information and values for the radiated VOR signal including bearing angle, modulation values, frequency values, and forward and reverse transmitter power values.

The DME information displayed consists of monitor and control panel alarm and status information, various timing and count values (such as system delay and pulse count), and transmitter power output.

The eight-point ground check contains measurements to check the accuracy of the radiated VOR signal.

The environmental readings indicate the conditions at this remote site. The signals include line frequency, line voltage, and indoor and outdoor temperatures.

Figure A-2 shows simulated information for a far field monitor/middle marker (FFM/MM) site, an inner marker, an outer marker, and for a Medium Intensity Approach Lighting System (MALSR). A message from a remote monitor channel to the concentrator is also displayed. The FFM/MM contains monitor and marker status information and values for the difference in depth of modulation (DDM) measured by the FFM.

Both the inner marker and outer marker contain the same kind of status information as the middle marker.

The MALSR contains status information for the steady burning lights and the flashing lights. The steady light voltage level setting, the voltage to the steady lights, and the flash rate per minute are also displayed.

Figure A-3 shows an eight-point ground check alarm that occurred because the measured 45° radial is out of tolerance (alarming value is 46.73°). The alarm data consists of the ground check values when the alarm occurred, the ground check values immediately before the alarm, and the latest measured ground check values.

FAC IDENT- 2

TIME: 13:03:38

VOR:

LATEST

MON A	/	NORMAL	/	ID BY-P	/	VAR	/	SUB CAR	/	IDENT	/
		ON		OFF		OFF		OFF		OFF	
MON B	/	NORMAL	/	ID BY-P	/	VAR	/	SUB CAR	/	IDENT	/
		ON		OFF		OFF		OFF		OFF	
CONTRL	/	TX1	/	TX1 SBY	/	TX2	/	TX2 SBY	/	FAIL	/
		ON		OFF		OFF		OFF		OFF	
RMM	/	BEARING	/	CRS WTH	/	CAR LVL	/	9960 MD	/	30 AM	/
		OFF		OFF		OFF		OFF		OFF	
RMM	/	BEARING	/	CRS WTH	/	CAR LVL	/	9960 MD	/	30 AM	/
		270.1		23.10		31.05		28.60		30.70	
RMM	/	1020 FQ	/	30 FM	/	F/9960 FQ	/				
		1019.		30.03		9960.					
POWER	/	FWD CAR	/	FWD SB1	/	FWD SB2	/	REV CAR	/	REV SB1	/
		100.0		2.154		1.080		2.414		.1441	

DME:

LATEST

MON A	/	SYS DLY	/	PWR OUT	/	RPY EFF	/	PUL SPC	/	PUL CNT	/	IDENT	/	BY-PASS	/
		OFF		OFF		OFF		OFF		OFF		OFF		OFF	
MON B	/	SYS DLY	/	PWR OUT	/	RPY EFF	/	PUL SPC	/	PUL CNT	/	IDENT	/	BY-PASS	/
		OFF		OFF		OFF		OFF		OFF		OFF		OFF	
CONTRL	/	SYS DLY	/	PWR OUT	/	RPY EFF	/	PUL SPC	/	PUL CNT	/	IDENT	/		
		OFF		OFF		OFF		OFF		OFF		OFF			
CONTRL	/	TX1	/	TX2	/	ANT 1	/	ANT 2	/	BY-PASS	/	SHUTDOWN	/	ALARM	/
		ON		OFF		ON		OFF		OFF		OFF		OFF	
MON A	/	SYS DLY	/	PUL WTH	/	RPY EFF	/	PUL SPC	/	PUL CNT	/	IDENT	/		
		49.40		3.501		65.87		12.00		2096.		1350.			
MON B	/	SYS DLY	/	PUL WTH	/	RPY EFF	/	PUL SPC	/	PUL CNT	/	IDENT	/		
		49.28		3.493		68.00		12.11		2095.		1350.			
POWER	/	PK TX1	/	PK TX2	/										
		1000.		.0000											

GROUND CHECK:

LATEST

/	0	/	45	/	90	/	135	/	180	/	225	/	270	/	315	/
	.0000		45.01		90.02		134.9		180.0		224.9		269.9		315.0	

ENV:

LATEST

/	LN FRER	/	LN VLTS	/	IN IMF	/	OUT IMF	/
	60.00		110.5		68.75		32.00	

82-89-A-1

FIGURE A-1. VOR/DME SIMULATED DATA

FAC IDENT- 4

TIME: 14:18:38

DATE: 03/25/82

FAR FIELD MONITOR

LATEST

/ FFM /SD ALRT/SHT DWN/MISMTCH/PWR-TMF/BYPASS/
ON OFF OFF OFF OFF OFF

MIDDLE MARKER

/ MAIN / STDBY / OFF /ABN MON/ ABN PE/ PERF /
ON OFF OFF OFF OFF OFF

/ DDM1 / DDM2 / DDM3 /
+0.01 -0.01 -0.04

FAC IDENT- 5

TIME: 14:18:46

DATE: 03/25/82

INNER MARKER

LATEST

/ MAIN / STDBY / OFF /ABN MON/ ABN PE/ PERF /
ON OFF OFF OFF OFF OFF

FAC IDENT- 6

TIME: 14:50:07

DATE: 03/25/82

OUTER MARKER

LATEST

/ MAIN / STDBY / OFF /ABN MON/ ABN PE/ PERF /
ON OFF OFF OFF OFF OFF

FAC IDENT- 7

TIME: 14:50:14

DATE: 03/25/82

MALSR

LATEST

/ STEADY/FLASHER/
ON OFF

/INT LVL/VOLTAGE/FLSH RT/
3 47 120

FAC IDENT- 4

This is a message from remote monitor #4 to the concentrator.

82-89-A-2

FIGURE A-2. ADDITIONAL SIMULATED DATA AND MESSAGES

FAC IDENT- 2

TIME: 15:00:33

GROUND CHECK:

ALARM

/	0	/	45	/	90	/	135	/	180	/	225	/	270	/	315	/
	.0000		46.73		90.02		134.9		180.0		224.9		269.9		315.0	

TIME: 15:00:32

GROUND CHECK:

PRE-ALARM

/	0	/	45	/	90	/	135	/	180	/	225	/	270	/	315	/
	.0000		45.01		90.02		134.9		180.0		224.9		269.9		315.0	

TIME: 15:00:35

GROUND CHECK:

LATEST

/	0	/	45	/	90	/	135	/	180	/	225	/	270	/	315	/
	.0000		46.73		90.02		134.9		180.0		224.9		269.9		315.0	

82-89-A-3

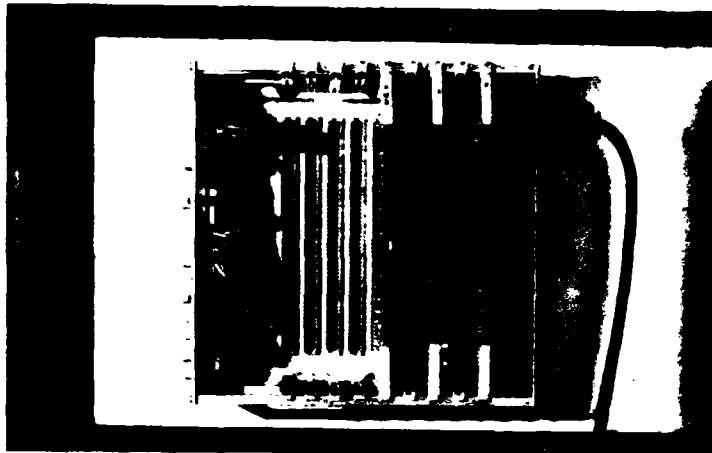
FIGURE A-3. GROUND CHECK SIMULATED ALARM DATA

APPENDIX B
SYSTEM PHOTOGRAPHS



FIGURE B-1. RMMS CONCENTRATOR

SIMULATOR # 4
FEM-MM LM OM MALSIR FACILITY #8,9,10,11



SIMULATOR # 1
VOR DME FACILITY # 1

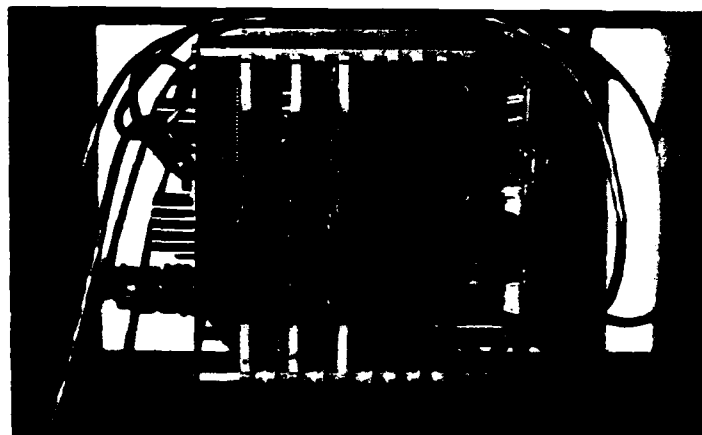


FIGURE B-2. REMOTE SITE SIMULATORS

— 8
DTIC